



TUGAS AKHIR - KI141502

SISTEM NAVIGASI *INDOOR* MENGGUNAKAN SINYAL WI-FI DAN KOMPAS DIGITAL BERBASIS INTEGRASI DENGAN *SMARTPHONE* UNTUK STUDI KASUS PADA GEDUNG BERTINGKAT

ALIFA RIDHO MUSTHAFA
NRP 5112100045

Dosen Pembimbing I
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. F.X. ARUNANTO, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

SISTEM NAVIGASI *INDOOR* MENGGUNAKAN SINYAL WI-FI DAN KOMPAS DIGITAL BERBASIS INTEGRASI DENGAN *SMARTPHONE* UNTUK STUDI KASUS PADA GEDUNG BERTINGKAT

ALIFA RIDHO MUSTHAFA
NRP 5112100045

Dosen Pembimbing I
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. F.X. ARUNANTO, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

INTEGRATION OF WI-FI AND DIGITAL COMPASS IN SMARTPHONES FOR INDOOR NAVIGATION SYSTEM IN MULTI-STOREY BUILDING

ALIFA RIDHO MUSTHAFA
NRP 5112100045

Supervisor I
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.

Supervisor II
Ir. F.X. ARUNANTO, M.Sc

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SISTEM NAVIGASI INDOOR MENGGUNAKAN SINYAL WI-FI DAN KOMPAS DIGITAL BERBASIS INTEGRASI DENGAN SMARTPHONE UNTUK STUDI KASUS PADA GEDUNG BERTINGKAT

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Algoritma dan Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

ALIFA RIDHO MUSTHAFA

NRP : 5112 100 045

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr.tech. Ir. R.V. Hari Ginard
NIP: 196505181992031003 (Pembimbing 1)
2. Ir. F.X. Arunanto, M.Sc.
NIP: 195701011983031004 (Pembimbing 2)

**SURABAYA
JUNI, 2016**

[Halaman ini sengaja dikosongkan]

SISTEM NAVIGASI *INDOOR* MENGGUNAKAN SINYAL WI-FI DAN KOMPAS DIGITAL BERBASIS INTEGRASI DENGAN *SMARTPHONE* UNTUK STUDI KASUS PADA GEDUNG BERTINGKAT

Nama Mahasiswa : Alifa Ridho Musthafa
NRP : 5112100045
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Ir. F.X. Arunanto, M.Sc

Abstrak

Saat ini ada banyak bangunan tinggi dan besar terutama di kota-kota besar. Bangunan tersebut tersusun dari banyak lantai, puluhan bahkan ratusan kamar dan koridor. Terkadang hal tersebut membuat seseorang merasa kesulitan dalam mencari dan menuju sebuah lokasi pada gedung bertingkat. Berkat teknologi Global Positioning System (GPS), navigasi saat ini sudah sangat maju untuk kasus di luar ruangan, tetapi belum ada standar khusus dan belum terlalu umum digunakan untuk kasus di dalam ruangan. Disisi lain, sudah ada sistem yang dapat menggantikan GPS untuk studi kasus di dalam ruangan, salah satunya yaitu sistem indoor localization berbasis Wi-Fi.

Dengan memanfaatkan sistem indoor localization berbasis Wi-Fi, dibuat sistem navigasi yang dapat digunakan di dalam ruangan. Sistem indoor localization digunakan untuk menentukan posisi pengguna. Setelah posisi pengguna diketahui, rute terbaik menuju tujuan pengguna ditentukan dengan bantuan algoritma A dan kompas digital digunakan untuk menentukan arah kemana pengguna harus pergi. Studi kasus ini akan dilakukan di kampus Teknik Informatika ITS.*

Secara umum, sistem indoor localization yang digunakan untuk menggantikan fungsi GPS dalam menentukan posisi memberikan performa yang baik. Sistem tersebut mampu menghasilkan presentase rata-rata akurasi pendeteksian lokasi sebesar 88,953% diambil dari seluruh kasus pada setiap lokasi yang dilewati dalam percobaan. Sedangkan untuk uji coba pencarian rute, sistem indoor navigation dapat memberikan rute terpendek pada semua kasus percobaan.

Kata kunci: Indoor Localization, Indoor Navigation, Wi-Fi, Kompas Digital, Algoritma A*

INTEGRATION OF WI-FI AND DIGITAL COMPASS IN SMARTPHONES FOR INDOOR NAVIGATION SYSTEM IN MULTI-STOREY BUILDING

Nama Mahasiswa : Alifa Ridho Musthafa
NRP : 5112100045
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Ir. F.X. Arunanto, M.Sc

Abstract

Nowadays there are many tall and large buildings, especially in big cities. The building is composed of many floors, tens or even hundreds of rooms and corridors. Sometimes someone feels difficulty in finding and toward a location in buildings. Thanks to the Global Positioning System (GPS) technology, navigation now is very advanced for outdoor case, but there are no specific standards and not too commonly used for indoor case. On the other side, there is a system that can replace GPS for indoor case, for example indoor localization system based on Wi-Fi.

By utilizing the indoor localization system based on Wi-Fi, indoor navigation system can be made. The indoor localization system is used to determine the user's position. After the user's position is known, the best route to the destination is determined with the A algorithm and digital compass is used to determine the direction the user should go. This case study is in Informatics Department Building.*

In general, the indoor localization system that is used to replace the function of GPS for positioning provide excellent performance. The system is able to generate average accuracy of location detection up to 88.185% in the experiment. Indoor

navigation system also provide best route in all experiments cases.

Keywords: Indoor Localization, Indoor Navigation, Wi-Fi, Digital Compass, A* Algorithm

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabil'alam, segala puji hanya milik Allah SubhanahuWata'alla, atas segala rahmat dan karunia-Nya yang tidak terhingga sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Sistem Navigasi Indoor Menggunakan Sinyal Wi-Fi dan Kompas Digital Berbasis Integrasi dengan Smartphone untuk Studi Kasus Pada Gedung Bertingkat”** dengan baik dan tepat waktu.

Melalui lembar ini, penulis hanya ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan kepada penulis selama ini.
2. Kedua orang tua penulis yang tiada henti mencurahkan kasih sayang, perhatian, doa, dan dukungan kepada penulis selama ini.
3. Ilham Ramadhan selaku saudara penulis yang memberikan doa dan dukungan kepada penulis dalam menuntut ilmu hingga saat ini.
4. Bapak Hari dan Bapak Arunanto selaku dosen pembimbing yang telah memberikan bimbingan, motivasi, dan meluangkan waktu untuk membantu pengerjaan Tugas Akhir ini.
5. Bapak dan Ibu dosen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.

6. Saudara Muhammad Faris Ghanianto dan Saudari Kharisma Alivia Nastiti yang memberikan bantuan dan semangat dalam pengerjaan Tugas Akhir ini.
7. Sahabat Tugas Akhir rumpun Algoritma Pemrograman yang turut berkontribusi dalam pengerjaan Tugas ini: Anggeriko, Metana, Fananda, Dinar, Ratih, dan Rina.
8. Sahabat Arkon yang selalu memberikan dukungan, inspirasi dan dukungan untuk terus berkarya: Sugiarto, Wimpy, dan Ibrahim.
9. Sahabat mahasiswa Klaten di Surabaya selaku teman seperjuangan.
10. Keluarga besar angkatan tercinta TC 2012 yang sudah menemani keseharian penulis di kampus perjuangan.
11. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu-persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2016

Alifa Ridho Musthafa

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Location Based Service (LBS)</i>	7
2.1.1 <i>Global Positioning System (GPS)</i>	7
2.1.2 <i>Indoor Positioning System (IPS)</i>	8
2.1.3 <i>Indoor Localization</i>	8
2.2 Gedung Teknik Informatika ITS	8
2.3 <i>Indoor Localization Menggunakan Sinyal Wi-Fi dan</i> <i>Clustering Filtered K-Nearest Neighbors</i>	9
2.4 Android Studio dan SDK Tools.....	11
2.5 SQLite	11
2.6 Lintasan Terpendek (<i>Shortest Path</i>)	11
2.7 <i>Graph</i>	12
2.8 Algoritma A*	13
2.9 Fungsi Heuristik	14
2.10 Kompas Digital Android	15
BAB III ANALISIS DAN PERANCANGAN	17
3.1 Deskripsi Umum Perangkat Lunak.....	17

3.1.1	Arsitektur Sistem	18
3.1.2	Kebutuhan Fungsional Aplikasi	20
3.2	Perancangan	25
3.2.1	Perancangan Basis Data.....	25
3.2.2	Perancangan Antarmuka Perangkat Lunak.....	27
3.2.3	Perancangan Proses Sistem Aplikasi <i>Indoor</i> <i>Navigation</i>	33
	BAB IV IMPLEMENTASI	41
4.1	Lingkungan Implementasi	41
4.1.1	Lingkungan Implementasi Perangkat Keras	41
4.1.2	Lingkungan Implementasi Perangkat Lunak	41
4.2	Implementasi Basis Data	42
4.2.1	Implementasi Struktur Basis Data	42
4.2.2	Implementasi <i>Query</i>	44
4.3	Implementasi Antarmuka.....	47
4.3.1	Antarmuka Halaman <i>Splash Screen</i>	47
4.3.2	Antarmuka Halaman Informasi Titik Awal dan Tujuan.....	48
4.3.3	Antarmuka Halaman <i>Edit Source</i>	49
4.3.4	Antarmuka Halaman <i>Edit Destination</i>	50
4.3.5	Antarmuka Halaman Navigasi.....	51
4.4	Implementasi Proses Sistem Aplikasi <i>Indoor</i> <i>Navigation</i>	52
4.4.1	Implementasi Prediksi Lokasi Pengguna	52
4.4.2	Implementasi Pencarian Rute Terpendek	53
4.4.3	Implementasi Menampilkan Informasi pada Peta	56
4.4.4	Implementasi Kompas Digital dan Panduan Arah.....	58
	BAB V UJI COBA DAN EVALUASI	61
5.1	Lingkungan Uji Coba	61
5.2	Dasar Pengujian	61
5.3	Pengujian Fungsionalitas	62
5.3.1	Pengujian Menampilkan Posisi Pengguna	62
5.3.2	Pengujian Melakukan Navigasi	63
5.4	Pengujian Akurasi.....	64
5.4.1	Skenario Pengujian Akurasi	65

5.4.2	Hasil Pengujian Akurasi.....	66
5.4.3	Evaluasi Hasil Pengujian.....	68
	BAB VI KESIMPULAN DAN SARAN.....	71
6.1	Kesimpulan.....	71
6.2	Saran.....	71
	DAFTAR PUSTAKA	73
	LAMPIRAN A - Pemodelan Lokasi dalam Koordinat Kartesian.....	75
	LAMPIRAN B – Gambar Peta.....	77
	LAMPIRAN C – Gambar Hasil Pengujian	79
	BIODATA PENULIS	81

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Gedung Teknik Informatika dari Google Maps ...	9
Gambar 2.2 Gedung Teknik Informatika dari Samping.....	9
Gambar 2.3 Arsitektur Sistem <i>Indoor Localization</i>	10
Gambar 3.1 Arsitektur Sistem.....	18
Gambar 3.2 Konsep <i>Web Service Indoor Localization</i>	19
Gambar 3.3 Diagram Kasus Penggunaan.....	21
Gambar 3.4 Diagram Aktivitas Kasus Penggunaan UC-01 ...	23
Gambar 3.5 Diagram Aktivitas Kasus Penggunaan UC-02 ...	24
Gambar 3.6 <i>Conceptual Data Model</i>	25
Gambar 3.7 <i>Physical Data Model</i>	25
Gambar 3.8 Rancangan Antarmuka Halaman <i>Splash Screen</i> ..	28
Gambar 3.9 Rancangan Antarmuka Halaman Informasi Titik Awal dan Tujuan	29
Gambar 3.10 Rancangan Antarmuka Halaman <i>Edit Source</i> ..	30
Gambar 3.11 Rancangan Antarmuka Halaman <i>Edit Destination</i>	31
Gambar 3.12 Rancangan Antarmuka Halaman Navigasi.....	32
Gambar 3.13 Pemberian Koordinat pada Denah Gedung Lantai 1	34
Gambar 3.14 Pemodelan Lokasi dalam <i>Graph</i>	35
Gambar 3.15 <i>Flowchart</i> Algoritma A*	37
Gambar 3.16 Tampilan Kompas	38
Gambar 3.17 Posisi Gedung terhadap Arah Mata Angin	38
Gambar 3.18 Peta Gedung Lantai 1	39
Gambar 3.19 <i>Marker</i> Posisi Pengguna.....	39
Gambar 3.20 Contoh Penggunaan <i>Marker</i> Posisi Pengguna ..	40
Gambar 3.21 <i>Marker</i> Tujuan Pengguna	40
Gambar 3.22 Contoh Penggunaan <i>Marker</i> Tujuan Pengguna ..	40
Gambar 4.1 Halaman <i>Splash Screen</i>	48
Gambar 4.2 Halaman Informasi Titik Awal dan Tujuan.....	49
Gambar 4.3 Halaman <i>Edit Source</i>	50
Gambar 4.4 Halaman <i>Edit Destination</i>	51
Gambar 4.5 Halaman Navigasi	52

Gambar 5.1 Pengujian Mengetahui Lokasi Pengguna.....	63
Gambar 5.2 Pengujian Melakukan Navigasi	65
Gambar 5.3 Hasil Pengujian pada Kasus 1.....	67
Gambar 5.4 Hasil Pengujian pada Kasus 2.....	67
Gambar 7.1 Pemodelan Lokasi dalam Koordinat Kartesius Lantai 2.....	75
Gambar 7.2 Pemodelan Lokasi dalam Koordinat Kartesius Lantai 3.....	76
Gambar 8.1 Gambar Peta Lantai 2	77
Gambar 8.2 Gambar Peta Lantai 3	78
Gambar 9.1 Hasil Pengujian pada Kasus 3.....	79
Gambar 9.2 Hasil Pengujian pada Kasus 4.....	79
Gambar 9.3 Hasil Pengujian pada Kasus 5.....	80
Gambar 9.4 Hasil Pengujian pada Kasus 6.....	80

DAFTAR TABEL

Tabel 3.1 Fungsi GetsPosition pada <i>Web Service</i>	19
Tabel 3.2 Deskripsi Kasus Penggunaan	21
Tabel 3.3 Rincian Kasus Penggunaan UC-01	22
Tabel 3.4 Rincian Kasus Penggunaan UC-02	23
Tabel 3.5 Detail Tabel Ruangan (<i>Room</i>).....	26
Tabel 3.6 Detail Tabel Tipe Lokasi (<i>LocationType</i>)	26
Tabel 3.7 Detail Tabel Posisi (<i>Position</i>)	27
Tabel 5.1 Spesifikasi Lingkungan Pengujian Perangkat Lunak	61
Tabel 5.2 Skenario Pengujian Mengetahui Lokasi Pengguna	62
Tabel 5.3 Skenario Pengujian Melakukan Navigasi	63
Tabel 5.4 Kasus Pengujian Akurasi	65

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Tabel Ruangan (<i>Room</i>).....	42
Kode Sumber 4.2 Implementasi Tabel Tipe Lokasi (<i>LocationType</i>).....	43
Kode Sumber 4.3 Implementasi Tabel Posisi (<i>Position</i>)	43
Kode Sumber 4.4 Mendapatkan Data Tipe Lokasi	44
Kode Sumber 4.5 Implementasi Mendapatkan Daftar Ruangan	45
Kode Sumber 4.6 Mendapatkan Data Ruangan	46
Kode Sumber 4.7 Mendapatkan Data Posisi	47
Kode Sumber 4.8 <i>Scanning</i> Sinyal WiFi	53
Kode Sumber 4.9 Pengiriman Hasil <i>Scanning</i> Sinyal WiFi...	53
Kode Sumber 4.10 Implementasi Fungsi <i>Manhattan Distance</i>	54
Kode Sumber 4.11 Fungsi Merekonstruksi <i>Path</i>	55
Kode Sumber 4.12 <i>Pseudocode</i> Implementasi Algoritma A*	56
Kode Sumber 4.13 Implementasi Merubah Warna Jalan pada Peta.....	56
Kode Sumber 4.14 Implementasi Menempatkan <i>Marker</i> Posisi Pengguna	57
Kode Sumber 4.15 Implementasi Menempatkan <i>Marker</i> Tujuan Pengguna	58
Kode Sumber 4.16 Aktifasi Sensor Orientasi	59
Kode Sumber 4.17 Implementasi Panduan Arah	60
Kode Sumber 4.18 Animasi Rotasi pada Gambar Tampilan Kompas	60

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan hal-hal yang menjadi latar belakang, permasalahan yang dihadapi, batasan masalah, tujuan dan manfaat, metodologi, dan sistematika penulisan yang digunakan dalam pembuatan Tugas Akhir.

1.1 Latar Belakang

Saat ini ada banyak bangunan tinggi dan besar terutama di kota-kota besar. Bangunan tersebut tersusun dari banyak lantai, puluhan bahkan ratusan kamar dan koridor. Terkadang hal tersebut membuat seseorang merasa kesulitan dalam mencari dan menuju sebuah lokasi pada gedung bertingkat. Sistem navigasi dapat membantu mencari dan menuju sebuah lokasi. Berkat teknologi Global Positioning System (GPS), navigasi saat ini sudah sangat maju untuk kasus di luar ruangan, tetapi belum ada standar khusus dan belum terlalu umum digunakan untuk kasus di dalam ruangan.

GPS mempunyai manfaat yang sangat besar pada sistem navigasi, namun sistem ini memiliki akurasi yang rendah saat pengguna berada di suatu ruangan atau bangunan [1]. GPS hanya dapat menunjukkan daerah atau nama jalan pada lokasi pengguna, akan tetapi sistem ini tidak dapat menunjukkan lokasi ruangan pada suatu bangunan dimana pengguna berada, apa nama dari ruangan tersebut dan lantai berapa lokasi ruangan tersebut. Oleh karena itu, diperlukan sistem lain yang lebih akurat ketika digunakan untuk navigasi di dalam ruangan.

Selain GPS, kompas adalah salah satu alat bantu dalam navigasi. Kompas mampu memberikan rujukan arah tertentu, sehingga sangat membantu dalam bidang navigasi. Seiring dengan kemajuan teknologi *smartphone*, kini kompas dapat dibuat pada perangkat *mobile* dengan *platform* Android [2].

Dalam Tugas Akhir ini dibuat sistem *indoor navigation* yang memanfaatkan sistem *indoor localization* menggunakan sinyal Wi-Fi. Sistem tersebut dapat menggantikan fungsi GPS sebagai

penentu posisi pengguna. Sistem *indoor localization* menggunakan sinyal Wi-Fi memiliki akurasi yang lebih baik ketika berada di dalam ruangan [3]. Setelah posisi pengguna diketahui, rute terbaik menuju lokasi tujuan pengguna akan dan ditampilkan pada peta. Kompas digital digunakan untuk menentukan arah kemana pengguna harus pergi. Studi kasus ini akan dilakukan di kampus Teknik Informatika ITS.

1.2 Rumusan Masalah

Berdasarkan uraian di atas dapat dirumuskan hal-hal berikut:

1. Bagaimana memanfaatkan sistem *indoor localization* menggunakan sinyal Wi-Fi sebagai penentu posisi pengguna pada sistem *indoor navigation*?
2. Bagaimana cara mencari rute terpendek pada sistem *indoor navigation*?
3. Bagaimana mengimplementasikan kompas digital pada sistem *indoor navigation*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini dibatasi oleh hal-hal sebagai berikut:

1. Lokasi gedung bertingkat berada di lingkungan Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember yaitu beberapa ruang kelas, plasa lantai 1 dan 2, ruang administrasi, serta beberapa ruang laboratorium di lantai 3.
2. Aplikasi dibangun khusus untuk perangkat bergerak dengan sistem operasi Android dengan versi minimum 4.2.
3. Data-data mengenai sinyal Wi-Fi didapat dengan memanfaatkan sensor penangkap sinyal Wi-Fi yang ada pada perangkat bergerak Android.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah membuat aplikasi *indoor navigation* dengan memanfaatkan sistem *indoor localization* menggunakan sinyal Wi-Fi dan kompas digital.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah memudahkan pengguna dalam mencari dan menuju sebuah lokasi pada pada gedung bertingkat dengan rute terpendek.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan yang sama dengan Tugas Akhir ini. Penyusunan proposal Tugas Akhir dilaksanakan untuk merumuskan masalah serta melakukan penetapan rancangan dasar dari sistem yang akan dikembangkan dalam pelaksanaan Tugas Akhir ini.

2. Studi literatur

Pada tahap ini dilakukan pemahaman informasi dan literatur yang diperlukan untuk tahap implementasi program. Tahap ini diperlukan untuk membantu memahami penggunaan komponen-komponen terkait dengan sistem yang akan dibangun, antara lain: sistem *indoor localization* menggunakan sinyal Wi-Fi, algoritma A*, dan kompas digital. Selain itu, juga dibantu beberapa literatur lain yang dapat menunjang proses penyelesaian Tugas Akhir ini.

3. Analisis dan perancangan perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat.

4. Implementasi perangkat lunak

Implementasi perangkat lunak merupakan tahap membangun rancangan program yang telah dibuat. Pada tahap

ini akan direalisasikan mengenai rancangan apa saja yang telah didefinisikan pada tahap sebelumnya. Implementasi ini akan menggunakan bahasa pemrograman Java.

5. Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba pada data yang telah dikumpulkan. Tahap ini digunakan untuk mengevaluasi kinerja program serta mencari masalah yang mungkin timbul saat program dievaluasi serta melakukan perbaikan jika terdapat kesalahan pada program.

6. Penyusunan buku Tugas Akhir

Pada tahap ini disusun buku yang memuat dokumentasi mengenai perancangan, pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu perumusan masalah, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Perangkat Lunak

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan Alur Sistem, Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada sistem.

Bab IV Implementasi

Bab ini membahas mengenai implementasi dari rancangan yang telah dibuat pada bab sebelumnya.

Penjelasan berupa *code* yang digunakan untuk proses implementasi dan juga sebagai penunjang fitur dalam aplikasi.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan mengenai kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari perangkat lunak yang telah dibuat sesuai dengan data yang diujikan.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

Daftar Pustaka

Merupakan daftar referensi yang digunakan dalam pembuatan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi hal-hal penting pada aplikasi.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori dan literatur yang menjadi dasar pembuatan Tugas Akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan.

2.1 *Location Based Service (LBS)*

Location Based Services (LBS) atau layanan berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu: *Geographic Information System*, *Internet Service*, dan *Mobile Device* [4]. Ada beberapa teknologi yang digunakan LBS untuk mendapatkan lokasi geografis dari sebuah perangkat *mobile*, diantaranya *Global Positioning System*, *Indoor Positioning System*, dan *Indoor Localization*. Salah satu contoh dari aplikasi yang menggunakan konsep LBS adalah aplikasi *maps*.

2.1.1 *Global Positioning System (GPS)*

GPS merupakan singkatan dari *Global Positioning System* yang merupakan sistem navigasi yang dapat memberikan informasi tentang sebuah lokasi. GPS merupakan teknologi yang awalnya digunakan untuk kepentingan militer dan sekarang dapat digunakan untuk kepentingan masyarakat biasa [5]. Posisi sebuah objek yang dideteksi oleh GPS biasanya direpresentasikan dalam bentuk koordinat garis bujur dan garis lintang. GPS dapat memberikan informasi lokasi dengan menggunakan minimal 3 satelit.

2.1.2 Indoor Positioning System (IPS)

Indoor Positioning System (IPS) adalah teknologi yang digunakan untuk mendeteksi posisi sebuah objek di dalam ruangan. Teknologi ini dapat menggantikan fungsi GPS ketika berada di dalam ruangan. Tidak seperti GPS yang memanfaatkan satelit, IPS memanfaatkan beberapa teknologi lain, diantaranya RFID, jaringan telpon seluler, sinar inframerah, Bluetooth, dan WLAN [5].

2.1.3 Indoor Localization

Indoor localization mempunyai kemiripan dengan *Indoor Positioning System* (IPS). Namun, ada perbedaan yang sangat jelas antara kata *positioning* dengan *localization*. *Indoor positioning* memiliki akurasi lebih detail dari pada teknologi *indoor localization*. *Indoor positioning* menentukan koordinat global dari sebuah lokasi (contoh: garis bujur dan garis lintang), sedangkan *indoor localization* menentukan koordinat relatif (contoh: Kampus Teknik Informatika, Ruang Administrasi) [5].

2.2 Gedung Teknik Informatika ITS

Gedung Teknik Informatika ITS terletak di Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya. Gedung ini mulai beroperasi pada tahun 2008 setelah sebelumnya jurusan Teknik Informatika menempati gedung lama yang berjarak 500 meter ke arah barat dari gedung saat ini [6].

Gedung Teknik Informatika memiliki 4 lantai dimana lantai 1 merupakan kelas untuk kuliah, lantai 2 merupakan ruang dosen dan aula, lantai 3 merupakan ruang laboratorium dan lantai 4 merupakan ruang *server*. Jika dilihat dari aplikasi Google Maps, gedung ini akan tampak seperti yang ditunjukkan oleh Gambar 2.1. Sedangkan jika dilihat dari samping gedung ini akan tampak seperti yang ditunjukkan oleh Gambar 2.2.



Gambar 2.1 Gedung Teknik Informatika dari Google Maps



Gambar 2.2 Gedung Teknik Informatika dari Samping

2.3 *Indoor Localization Menggunakan Sinyal Wi-Fi dan Clustering Filtered K-Nearest Neighbors*

Sistem *indoor localization* yang memiliki kemampuan untuk mendeteksi lokasi pengguna di dalam ruangan dengan menggunakan data kekuatan sinyal Wi-Fi yang ditangkap oleh

smartphone pada ruangan tersebut. Hasil yang didapatkan adalah informasi lokasi pengguna seperti: nama ruangan atau area lokasi keberadaan pengguna serta tingkat lantai dari lokasi tersebut.

Sistem ini dikembangkan dengan menggunakan bahasa pemrograman C# pada ASP.NET *Web Service*. Untuk dapat mendeteksi lokasi pengguna, sistem ini mengolah data sinyal Wi-Fi yang tertangkap oleh *smartphone* kemudian mengolah data tersebut dengan menggunakan algoritma *Clustering Filtered k-NN*. Arsitektur sistem ini ditunjukkan oleh Gambar 2.3.



Gambar 2.3 Arsitektur Sistem *Indoor Localization*

Sistem ini sudah dikembangkan pada studi kasus pada Gedung Teknik Informatika ITS. Sistem memberikan performa yang baik dengan persentase rata-rata akurasi pendeteksian lokasi sebesar 93,21% untuk seluruh *test area* pada setiap lokasi uji coba [3]. Sistem ini dapat menggantikan fungsi GPS sebagai penentu posisi pengguna ketika berada di dalam ruangan.

2.4 Android Studio dan SDK Tools

Android Studio adalah sebuah lingkungan pengembangan resmi yang digunakan untuk mengembangkan perangkat lunak berbasis sistem operasi Android. Android Studio dapat berjalan di berbagai *platform*, seperti Windows, Linux dan Macintosh. Spesifikasi *hardware* minimal yang diharuskan agar Android Studio dapat berjalan adalah memiliki 2 GB RAM, memiliki sisa penyimpanan sebesar 2 GB, dan telah terpasang Java Development Kit [7].

Perangkat lunak berbasis Android ditulis dalam bahasa pemrograman Java. Android SDK Tools digunakan untuk menyusun *script* yang ditulis menjadi sebuah perangkat lunak berbasis Android atau yang bisa disebut APK (*Android Package*).

2.5 SQLite

SQLite adalah sebuah *open source database* sangat terkenal pada perangkat kecil seperti Android karena cukup stabil. Pada *platform* Android, SQLite dijadikan satu di dalam Android *runtime*, sehingga setiap aplikasi Android dapat membuat basis data SQLite. SQLite menggunakan antarmuka SQL, sehingga cukup mudah untuk digunakan.

Terdapat beberapa alasan mengapa SQLite sangat cocok untuk pengembangan aplikasi Android, yaitu:

- *Database* tanpa konfigurasi. Hal ini membuatnya relatif mudah digunakan.
- Tidak membutuhkan *server*. Tidak ada proses *database* SQLite yang berjalan. Pada dasarnya satu set *libraries* telah menyediakan fungsionalitas *database*.
- *Single file database*. Hal ini membuat SQLite memiliki tingkat keamanan yang baik.
- *Open source*. Hal ini memudahkan pengembang dalam mengembangkan aplikasi [8].

2.6 Lintasan Terpendek (*Shortest Path*)

Lintasan Terpendek (*Shortest Path*) adalah lintasan minimum yang diperlukan untuk mencapai suatu titik dari titik

tertentu. Dalam pencarian lintasan terpendek, masalah yang dihadapi adalah mencari lintasan mana yang akan dilalui sehingga didapat lintasan yang paling pendek dari satu *verteks* ke *verteks* yang lain [9].

Ada beberapa macam persoalan lintasan terpendek, antara lain:

1. Lintasan terpendek antara dua buah *verteks*.
2. Lintasan terpendek antara semua pasangan *verteks*.
3. Lintasan terpendek dari *verteks* tertentu ke semua *verteks* yang lain.
4. Lintasan terpendek antara dua buah *verteks* yang melalui beberapa *verteks* tertentu.

2.7 *Graph*

Graph merupakan pasangan himpunan (V, E) ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul. *Graph* yang hanya mempunyai satu buah simpul tanpa sebuah sisi (*edges*) dinamakan *trivial graph* [10].

Sisi pada *graph* dapat mempunyai orientasi arah [11]. Berdasarkan orientasi arah pada sisi, maka secara umum *graph* dibedakan atas 2 jenis yaitu:

1. *Graph* tidak berarah (*undirected graph*)
Graph yang sisinya tidak mempunyai orientasi arah disebut *graph* tidak berarah. Pada *graph* tidak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u, v) = (v, u)$ adalah sisi yang sama.
2. *Graph* berarah (*directed graph* atau *digraph*)
Graph yang sisinya diberikan orientasi arah disebut sebagai *graph* berarah atau sisi berarah. Pada *graph* berarah, (u, v) dan (v, u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u, v) \neq (v, u)$. Untuk busur (u, v) , simpul u dinamakan simpul asal (*initial vertex*) dan simpul v dinamakan simpul terminal (*terminal vertex*).

2.8 Algoritma A*

Algoritma A* digunakan dalam menemukan lintasan terpendek pada suatu graf dan merupakan pengembangan dari algoritma *Best First Search* (BFS). Sama seperti BFS, algoritma A* juga menggunakan fungsi heuristik. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi matematika dituliskan sebagai berikut:

$$f(n) = g(n) + h(n) \quad (2.1)$$

Keterangan:

- $f(n)$ = Estimasi solusi *cost* terkecil melewati *node n*.
- $g(n)$ = *Cost* dari *start node* untuk mencapai *node n*.
- $h(n)$ = Estimasi *cost* terkecil dari *node n* ke tujuan.

Dengan perhitungan biaya seperti ini, algoritma A* adalah algoritma yang *complete* dan *optimal*. Hal ini berbeda dengan *Greedy best-first search* yang hanya memperhitungkan biaya perkiraan saja. Dengan biaya perkiraan yang baik, algoritma ini dapat menyajikan solusi yang optimal [12]. Algoritma A* secara ringkas langkah demi langkahnya adalah sebagai berikut:

1. Tambahkan *starting point* ke dalam *openset*.
2. Ulangi langkah berikut:
 - a. Mencari biaya F terendah pada setiap simpul dalam *open set*. Node dengan biaya F terendah kemudian disebut *current node*.
 - b. Memasukkan *current node* ke dalam *closed set*.
 - c. Untuk simpul (*neighbor node*) yang berdekatan dengan *current node*:
 - Jika termasuk *closed set*, maka abaikan.
 - Jika tidak ada pada *open set*, tambahkan ke *open set*.
 - jika sudah ada pada *open set*, periksa apakah ini jalan dari simpul ini ke *current node* yang lebih baik dengan menggunakan biaya G sebagai ukurannya. Simpul dengan biaya G yang lebih rendah berarti

bahwa ini adalah jalan yang lebih baik. Jika demikian, membuat simpul ini (*neighbor node*) sebagai *came from* dari *current node*, dan menghitung ulang nilai G dan F dari simpul ini.

d. Berhenti ketika:

- menambahkan target *point* ke dalam *closed set*, dalam hal ini jalan telah ditemukan, atau
 - gagal untuk menemukan target *point*, dan *open set* kosong. Dalam kasus ini tidak ada solusi.
3. Simpan jalan. Bekerja mundur dari target *point*, pergi dari masing-masing simpul ke simpul *came from* sampai mencapai *starting point*.

2.9 Fungsi Heuristik

Dalam metode pencarian heuristik, digunakan suatu fungsi heuristik yang digunakan untuk mengevaluasi keadaan-keadaan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan. Suatu fungsi dapat diterima sebagai fungsi heuristik jika biaya perkiraan yang dihasilkan tidak melebihi dari biaya sebenarnya. Suatu fungsi heuristik dapat dikatakan sebagai fungsi heuristik yang baik, apabila dapat memberikan biaya perkiraan yang mendekati biaya sebenarnya. Semakin mendekati biaya sebenarnya, fungsi heuristik tersebut semakin baik [12].

Dalam masalah pencarian rute terpendek, salah satu fungsi heuristik yang dapat digunakan adalah *Manhattan distance*. Fungsi heuristik ini akan menghitung jarak dari suatu titik menuju titik lainnya di bidang *cartesian* dengan menyusuri bagian vertikal dan horizontal tanpa pernah kembali. Secara sederhana sama dengan jumlah dari selisih absis dan selisih ordinat. Fungsi *Manhattan distance* dalam notasi matematika dituliskan sebagai berikut:

$$d(xyz) = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| \quad (2.2)$$

2.10 Kompas Digital Android

Kompas digital di ponsel dan tablet Android diaktifkan oleh sebuah sensor yang disebut sensor magnet. Sensor tersebut digunakan untuk mengukur kekuatan dan arah medan magnet. Dengan menganalisis medan magnet bumi, sensor ponsel dapat menentukan orientasi yang cukup akurat [2].

Kompas digital di Android juga bisa dibuat dengan menggunakan *orientation sensor*. *Orientation sensor* merupakan gabungan dari *geomagnetic sensor* dan *accelerometer*. Sensor tersebut dapat memberi data berupa sudut antara kutub utara magnet bumi dan sumbu y dari perangkat (*Azimuth*). Jika sumbu y dari perangkat sejajar dengan kutub utara magnet bumi maka nilai *Azimuth* adalah nol [13].

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN

Pada proses pembuatan perangkat lunak tidak terlepas dengan adanya proses perancangan. Pada bab ini akan dijelaskan perancangan sistem yang dibuat pada Tugas Akhir ini. Penjelasan tersebut meliputi deskripsi mengenai perangkat lunak, arsitektur sistem, dan alur berjalannya sistem.

3.1 Deskripsi Umum Perangkat Lunak

Aplikasi yang dibangun pada Tugas Akhir ini yaitu suatu aplikasi perangkat bergerak Android yang berbasis *Location Base Service*. Aplikasi ini memiliki kemampuan untuk menentukan posisi, rute terbaik, dan arah perjalanan untuk mencapai lokasi tujuan pengguna. Semua informasi tersebut akan disajikan pada layar *smartphone* pengguna.

Untuk menjalankan fungsinya, aplikasi ini melalui tiga tahapan. Tahap yang pertama adalah menentukan posisi lokasi dan tujuan pengguna. Penentuan posisi pada sistem ini dilakukan dengan memanfaatkan sistem *indoor localization* menggunakan sinyal Wi-Fi dan *clustering filtered K-Nearest Neighbors*. Sistem tersebut telah dikembangkan dalam *platform web service*. Untuk dapat menentukan lokasi, pengguna mengirimkan data RSS yang terdeteksi pada posisinya. Data RSS berupa beberapa BSSID (*Basic Service Set Identifier*) beserta kekuatan sinyal yang tertangkap. Data tersebut dikirimkan ke *server* untuk diproses menjadi informasi lokasi pengguna.

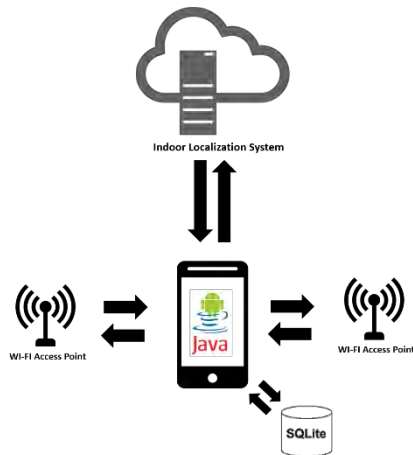
Setelah posisi dan tujuan pengguna diketahui, tahap selanjutnya adalah pencarian rute. Pencarian rute dalam sistem ini menggunakan algoritma A*. Rute terbaik akan ditampilkan pada sebuah peta pada *smartphone* pengguna. Tahap terakhir yakni menampilkan panduan arah yang dibantu dengan kompas digital.

Aplikasi *indoor navigation* ini dikembangkan dalam *platform* Android. Bahasa pemrograman yang digunakan adalah

Java. Untuk menyimpan segala informasi yang dibutuhkan dalam sistem ini, digunakan SQLite sebagai basis data.

3.1.1 Arsitektur Sistem

Arsitektur sistem pada aplikasi Tugas Akhir ini dijelaskan pada Gambar 3.1.



Gambar 3.1 Arsitektur Sistem

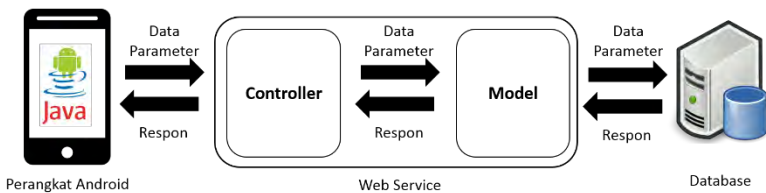
Sistem navigasi ini menggunakan arsitektur *client server*. *Server* merupakan *web service* dan *database server*, sedangkan *client* merupakan aplikasi pada *smartphone* berbasis Android. *Server* memanfaatkan sistem yang sudah ada sebelumnya, yakni sistem *indoor localization* menggunakan sinyal Wi-Fi dan *clustering filtered k-nearest neighbors*. Sistem tersebut digunakan untuk mengetahui posisi pengguna saat melakukan navigasi. *Client* bertugas mengumpulkan data, mengirimkan data ke *server*, mencari rute terbaik, dan menampilkan segala informasi terkait navigasi.

Berdasarkan perancangan arsitektur umum sistem pada Gambar 3.1, informasi lokasi pengguna dapat diketahui dengan mendeteksi data sinyal Wi-Fi yang ditangkap oleh *client* pada saat

itu. Kemudian *client* mengirimkan data tersebut ke *server*. Informasi lokasi pengguna kemudian digunakan *client* sebagai titik awal pada pencarian rute menuju tujuan.

3.1.1.1 Arsitektur Sistem Web Service Indoor Localization

Pada bagian ini dijelaskan mengenai arsitektur *web service indoor localization* yang digunakan untuk menentukan posisi pada sistem *indoor navigation*. Sistem *indoor localization* yang digunakan dikembangkan dengan menggunakan bahasa pemrograman C# pada ASP.NET Web Service. Konsep dari *web service* ditunjukkan pada Gambar 3.2. Pada Gambar 3.2, data dari perangkat Android akan masuk ke dalam *controller*. Data tersebut kemudian dibandingkan dengan data yang disimpan pada *database* melalui *model* sehingga data posisi pengguna dapat diketahui.



Gambar 3.2 Konsep Web Service Indoor Localization

Sistem *web service indoor localization* ini memiliki sebuah fungsi *GetPosition* yang digunakan untuk menentukan posisi pengguna. Penjelasan mengenai fungsi tersebut dijelaskan pada Tabel 3.1.

Tabel 3.1 Fungsi GetPosition pada Web Service

Operasi	Isi Operasi	Penjelasan
Metode	POST	Data parameter dikirim dengan menggunakan metode POST.

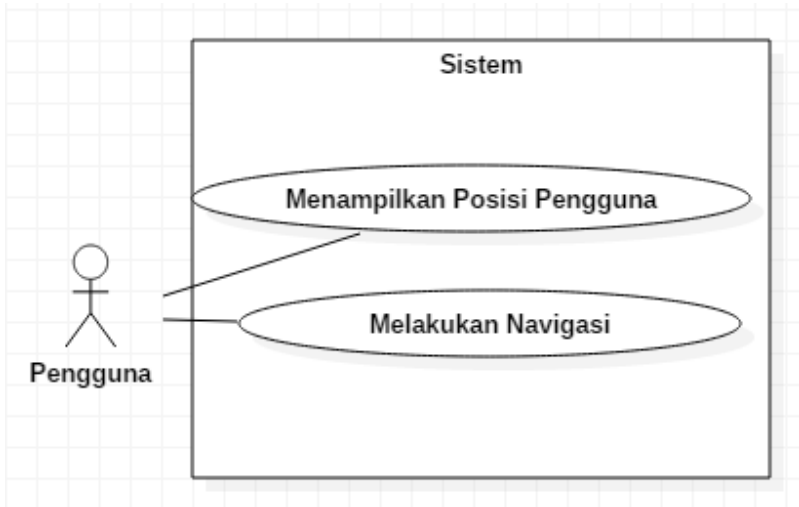
Operasi	Isi Operasi	Penjelasan
<i>Request</i>	http://wifiindoornavigation.azurewebsites.net/IPS_Services.asmx?op=GetPosition	Nilai dari data parameter akan dikirim ke alamat tersebut untuk diolah lebih lanjut.
Parameter	<i>datas</i>	Informasi yang dikirimkan berupa data BSSID dan kekuatan sinyal dari Wi-Fi di sekitar pengguna. Kedua data tersebut digabung membentuk sebuah <i>string</i> .
Respon/Hasil	<string xmlns="http://loki.com/">STUDIO</string>	Apabila posisi pengguna diketahui, sistem akan mengembalikan data berupa kode unik ruangan atau lokasi.
	<string xmlns="http://loki.com/">DISLOCATION</string>	Apabila posisi pengguna tidak diketahui, sistem akan mengembalikan data teks “ <i>DISLOCATION</i> ”.

3.1.2 Kebutuhan Fungsional Aplikasi

Berikut daftar kebutuhan fungsional dari aplikasi berbasis perangkat bergerak Android:

1. Menampilkan posisi pengguna
2. Melakukan navigasi

Kebutuhan fungsional aplikasi berbasis perangkat bergerak Android ini juga ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram Kasus Penggunaan

Diagram kasus penggunaan di atas akan dijelaskan lebih lengkap pada Tabel 3.2.

Tabel 3.2 Deskripsi Kasus Penggunaan

No	Kode	Nama Diagram Kasus	Keterangan
1	UC-01	Menampilkan Posisi Pengguna	Pengguna dapat mengetahui posisinya saat ini
2	UC-02	Melakukan Navigasi	Pengguna dapat mengetahui rute dan panduan arah dari titik awal menuju sebuah lokasi

3.1.2.1 Deskripsi Kasus Penggunaan UC-01

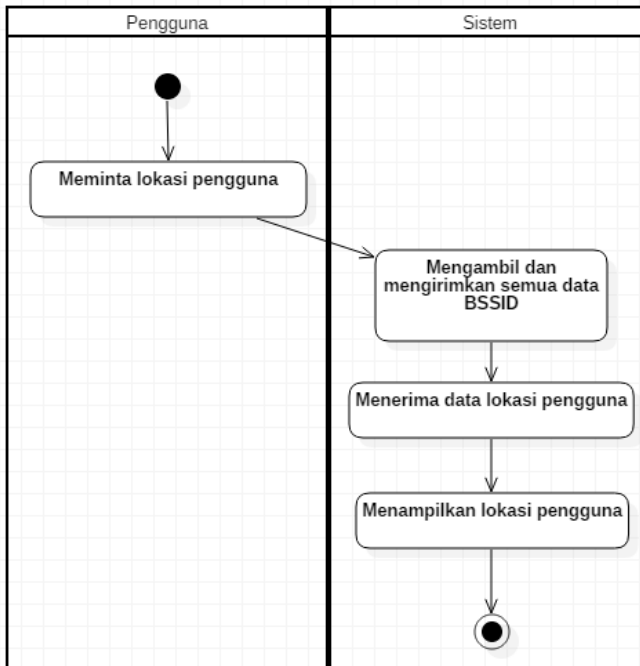
Kasus Penggunaan UC-01 mengenai mengetahui lokasi pengguna merupakan kasus penggunaan bagi pengguna yang ingin

mengetahui lokasi keberadaannya. Kasus penggunaan ini dijelaskan pada Tabel 3.3 dan juga diagram aktivitas pada Gambar 3.4.

Ketika sistem menampilkan posisi pengguna, sistem membutuhkan data BSSID beserta kekuatan sinyalnya. Data tersebut kemudian akan dikirim ke *server* sistem *indoor localization* dan diolah menjadi informasi posisi pengguna berada. Setelah posisi pengguna diketahui, informasi tentang posisi pengguna selanjutnya akan dikirimkan ke aplikasi *client*. Aplikasi *client* selanjutnya menampilkan informasi posisi pengguna berada pada layar *smartphone*.

Tabel 3.3 Rincian Kasus Penggunaan UC-01

Nama Kasus Penggunaan	Menampilkan Posisi Pengguna
Nomor	UC-01
Aktor	Pengguna
Kondisi Awal	Sistem belum menampilkan posisi pengguna berada
Kondisi Akhir	Sistem menampilkan posisi pengguna berada
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan menampilkan lokasi pengguna berada 2. Sistem mengambil dan mengirimkan semua data BSSID beserta kekuatan sinyalnya di sekitar posisi pengguna ke <i>server indoor localization</i> 3. Sistem menerima informasi lokasi pengguna dari <i>server indoor localization</i> 4. Sistem menampilkan posisi pengguna pada layar aplikasi <i>client</i>



Gambar 3.4 Diagram Aktivitas Kasus Penggunaan UC-01

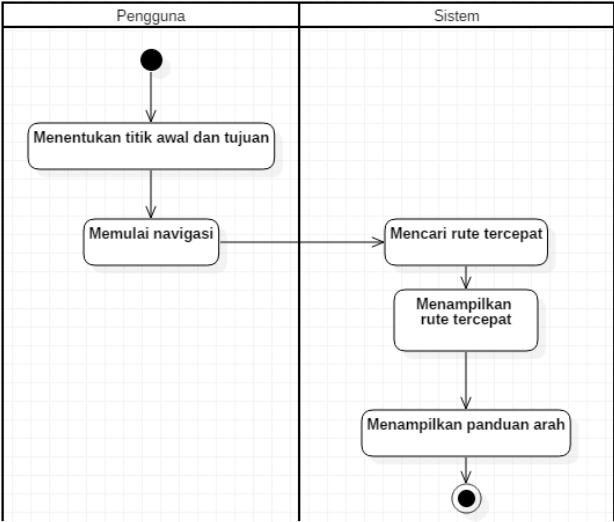
3.1.2.2 Deskripsi Kasus Penggunaan UC-02

Kasus Penggunaan UC-02 mengenai melakukan navigasi merupakan kasus penggunaan bagi pengguna yang meminta rute terpendek menuju sebuah lokasi beserta panduan arahnya. Kasus penggunaan ini dijelaskan pada Tabel 3.4 dan juga diagram aktivitas pada Gambar 3.5.

Tabel 3.4 Rincian Kasus Penggunaan UC-02

Nama Kasus Penggunaan	Melakukan Navigasi
Nomor	UC-02
Aktor	Pengguna

Kondisi Awal	Sistem belum menampilkan rute dan panduan arah menuju tujuan
Kondisi Akhir	Sistem menampilkan rute dan panduan arah menuju tujuan
Alur Normal	<ol style="list-style-type: none">1. Pengguna menentukan titik awal dan tujuan2. Pengguna menekan tombol navigasi3. Sistem mencari rute berdasarkan titik awal dan tujuan4. Sistem menampilkan rute menuju tujuan5. Sistem menampilkan panduan arah



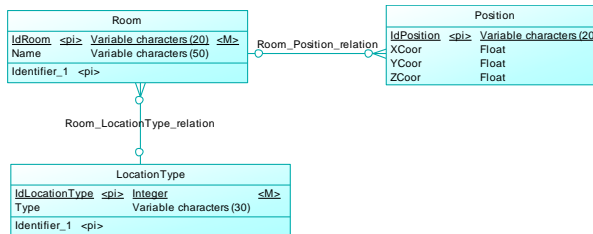
Gambar 3.5 Diagram Aktivitas Kasus Penggunaan UC-02

3.2 Perancangan

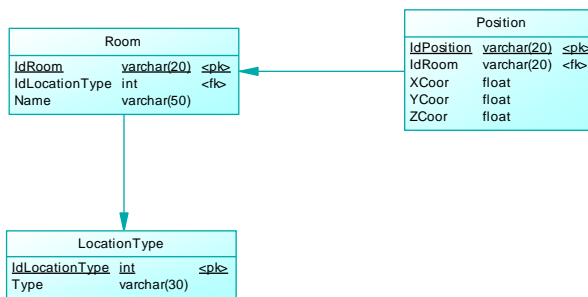
Pada bagian ini dijelaskan mengenai perancangan dari perangkat lunak *indoor navigation*, yang meliputi perancangan basis data, perancangan antarmuka perangkat lunak dan perancangan proses sistem *indoor navigation*. Pembahasan lebih detail akan dibahas berikut ini.

3.2.1 Perancangan Basis Data

Pada bagian ini akan dijelaskan mengenai rancangan basis data yang digunakan pada sistem aplikasi *indoor navigation*. Basis data yang dibangun pada sistem ini menggunakan manajemen basis data SQLite. Basis data tersebut digunakan untuk menyimpan data ruangan, jenis ruangan, dan posisi. Rancangan basis data berupa *Conceptual Data Model* dan *Physical Data Model* dapat dilihat pada Gambar 3.6 dan Gambar 3.7.



Gambar 3.6 Conceptual Data Model



Gambar 3.7 Physical Data Model

3.2.1.1 Rancangan Tabel Ruangan (*Room*)

Tabel ruangan digunakan untuk menyimpan data-data ruangan atau lokasi. Data yang disimpan berupa nama dari ruangan. Rancangan tabel ruangan dijelaskan pada Tabel 3.5.

Tabel 3.5 Detail Tabel Ruangan (*Room*)

Nama Kolom	Tipe Data	Keterangan
idRoom	<i>Varchar(20)</i>	<i>Primary key</i> untuk tabel <i>Room</i>
idLocationType	<i>integer</i>	<i>Foreign key</i> dari tabel <i>LocationType</i>
Name	<i>Varchar(50)</i>	Nama lengkap dari ruangan

Tabel ruangan memiliki relasi ke tabel lainnya yakni sebagai berikut.

1. Tabel Posisi (*Position*)
Tabel ruangan menyimpan informasi ruangan lengkap untuk setiap posisi dalam tabel posisi. Setiap ruangan memiliki beberapa posisi.
2. Tabel Tipe Lokasi (*LocationType*)
Tabel tipe lokasi menyimpan informasi jenis dari sebuah lokasi atau ruangan. Setiap tipe lokasi dimiliki oleh banyak ruangan atau lokasi.

3.2.1.2 Rancangan Tabel Tipe Lokasi (*LocationType*)

Tabel tipe lokasi digunakan untuk menyimpan data tipe lokasi dari sebuah ruangan atau lokasi. Rancangan tabel tipe lokasi dijelaskan pada Tabel 3.6.

Tabel 3.6 Detail Tabel Tipe Lokasi (*LocationType*)

Nama Kolom	Tipe Data	Keterangan
idLocationType	<i>integer</i>	<i>Primary key</i> untuk tabel <i>LocationType</i>
type	<i>Varchar(30)</i>	Tipe atau jenis lokasi

Tabel tipe lokasi ini memiliki relasi dengan tabel ruangan (*Room*). Dimana setiap tipe lokasi dimiliki oleh banyak ruangan.

3.2.1.3 Rancangan Posisi (*Position*)

Tabel posisi digunakan untuk menyimpan semua posisi yang terdapat pada lokasi gedung. Setiap lokasi pada suatu ruangan, akan dibagi setiap ukuran $1,2\text{m} \times 1,2\text{m}$. Posisi pada setiap ukuran tersebut akan disimpan di tabel posisi. Rancangan tabel posisi dijelaskan pada Tabel 3.7.

Tabel 3.7 Detail Tabel Posisi (*Position*)

Nama Kolom	Tipe Data	Keterangan
IdPosition	<i>Varchar(20)</i>	<i>Primary key</i> untuk tabel <i>Position</i>
IdRoom	<i>Varchar(20)</i>	<i>Foreign key</i> dari tabel <i>Room</i>
XCoor	<i>Float</i>	Koordinat X pada posisi
YCoor	<i>Float</i>	Koordinat Y pada posisi
ZCoor	<i>Float</i>	Koordinat Z pada posisi

Tabel posisi memiliki relasi dengan tabel ruangan (*Room*). Hubungan dengan tabel ruangan adalah tabel ruangan akan menyediakan informasi nama ruangan pada setiap posisi.

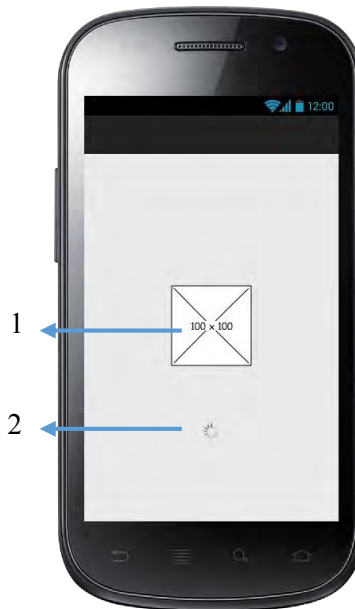
3.2.2 Perancangan Antarmuka Perangkat Lunak

Pada bagian ini dijelaskan mengenai rancangan antarmuka perangkat lunak yang akan dibuat. Antarmuka ini dirancang untuk *smartphone* berbasis Android.

3.2.2.1 Antarmuka Halaman *Splash Screen*

Halaman *Splash Screen* merupakan antarmuka yang pertama kali ditampilkan ketika aplikasi dijalankan. Pada antarmuka ini, pertama kali akan dilakukan pengecekan koneksi Wi-Fi atau koneksi data, bila tidak ditemukan maka aplikasi akan menampilkan notifikasi dan secara otomatis keluar dari aplikasi.

Kemudian apabila koneksi data tersedia, akan dilakukan proses menghidupkan jalur koneksi Wi-Fi jika sebelumnya belum diaktifkan. Setelah semua proses selesai dilakukan, aplikasi akan menuju halaman informasi titik awal dan tujuan. Rancangan halaman *splash screen* ditunjukkan pada Gambar 3.8.



Gambar 3.8 Rancangan Antarmuka Halaman *Splash Screen*

Berikut penjelasan antarmuka halaman *splash screen* berdasarkan nomor yang terdapat pada Gambar 3.8:

1. Gambar logo aplikasi.
2. Animasi *circular progress*.

3.2.2.2 Antarmuka Halaman Informasi Titik Awal dan Tujuan

Halaman informasi titik awal (*source*) dan tujuan (*destination*) digunakan untuk menampilkan informasi tentang titik awal dan tujuan navigasi pengguna. Pada halaman ini terdapat

pilihan untuk mengubah titik awal dan tujuan navigasi pengguna. Antarmuka halaman informasi titik awal dan tujuan ditunjukkan pada Gambar 3.9.



Gambar 3.9 Rancangan Antarmuka Halaman Informasi Titik Awal dan Tujuan

Berikut penjelasan antarmuka halaman informasi titik awal dan tujuan berdasarkan nomor yang terdapat pada Gambar 3.9:

1. Label bagian *source*.
2. Label *Position* bagian *source*.
3. Label untuk nama lokasi atau ruangan bagian *source*.
4. Label *Floor* bagian *source*.
5. Label untuk letak lantai bagian *source*.
6. Tombol untuk menuju halaman *edit source*.
7. Tombol untuk menentukan *source* secara otomatis.

8. Label bagian *destination*.
9. Label *Position* bagian *destination*.
10. Label untuk nama lokasi atau ruangan bagian *destination*.
11. Label *Floor* bagian *destination*.
12. Label untuk letak lantai bagian *destination*.
13. Tombol untuk menuju halaman *edit destination*.
14. Tombol untuk memulai navigasi.

3.2.2.3 Antarmuka Halaman *Edit Source*

Halaman *edit source* digunakan oleh pengguna untuk menentukan titik awal navigasi. Pada halaman tersebut pengguna memilih lokasi-lokasi yang tersedia sebagai titik awal. Antarmuka halaman *edit source* ditunjukkan pada Gambar 3.10.



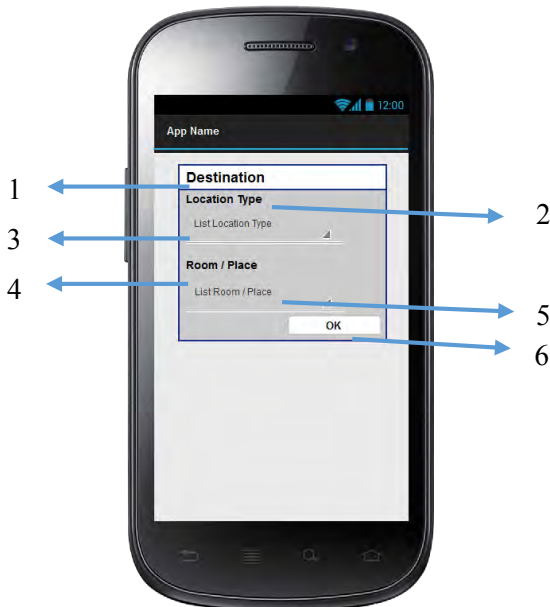
Gambar 3.10 Rancangan Antarmuka Halaman *Edit Source*

Berikut penjelasan antarmuka halaman *edit source* berdasarkan nomor yang terdapat pada Gambar 3.10:

1. Label *source*.
2. Label *Location Type*.
3. *Spinner* untuk menampilkan jenis atau tipe lokasi.
4. Label *Room / Place*.
5. *Spinner* untuk menampilkan nama ruangan atau lokasi.
6. Tombol memilih lokasi titik awal.

3.2.2.4 Antarmuka Halaman *Edit Destination*

Halaman *Edit Destination* digunakan oleh pengguna untuk menentukan tujuan navigasi. Pada halaman tersebut pengguna memilih lokasi-lokasi yang tersedia sebagai tujuan. Antarmuka halaman *edit destination* ditunjukkan pada Gambar 3.11.



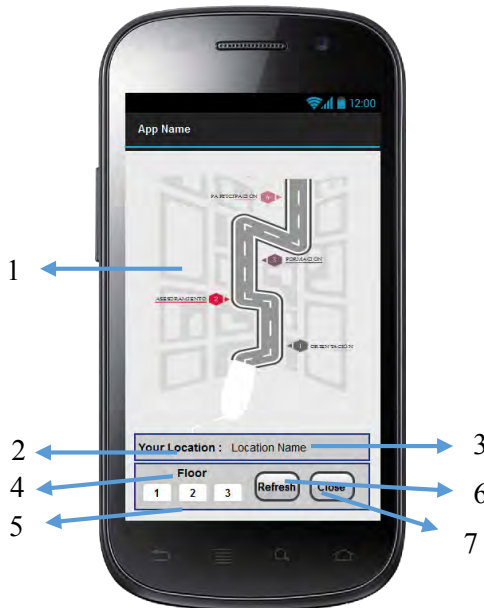
Gambar 3.11 Rancangan Antarmuka Halaman *Edit Destination*

Berikut penjelasan antarmuka halaman *edit destination* berdasarkan nomor yang terdapat pada Gambar 3.11.

1. Label *destination*.
2. Label *Location Type*.
3. *Spinner* untuk menampilkan jenis atau tipe lokasi.
4. Label *Room / Place*.
5. *Spinner* untuk menampilkan daftar nama ruangan atau lokasi.
6. Tombol memilih lokasi tujuan.

3.2.2.5 Antarmuka Halaman Navigasi

Halaman navigasi digunakan untuk memandu pengguna menuju tujuan. Pada halaman tersebut, terdapat peta yang menampilkan posisi pengguna, tujuan pengguna serta jalan atau rute menuju tujuan. Antarmuka halaman navigasi ditunjukkan pada Gambar 3.12.



Gambar 3.12 Rancangan Antarmuka Halaman Navigasi

Berikut penjelasan antarmuka halaman posisi pengguna berdasarkan nomor yang terdapat pada Gambar 3.12:

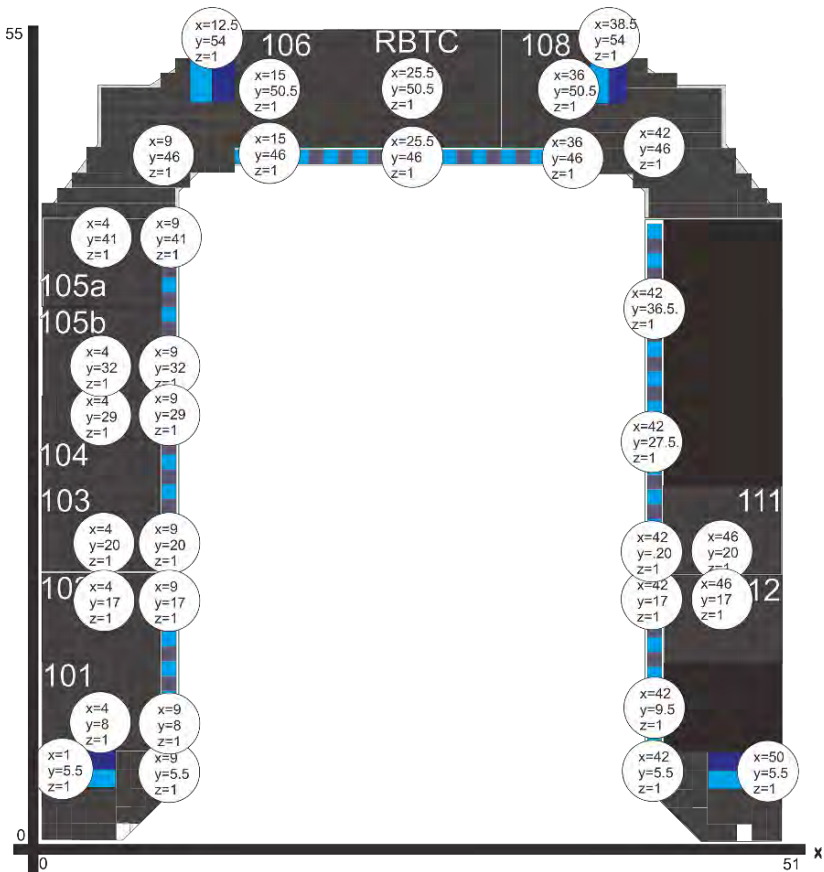
1. Gambar peta.
2. Label lokasi pengguna.
3. *Text* informasi lokasi pengguna.
4. Label *Floor*.
5. Tombol pilihan lantai peta.
6. Tombol *update* lokasi untuk memperbarui posisi pengguna.
7. Tombol untuk menghentikan navigasi.

3.2.3 Perancangan Proses Sistem Aplikasi *Indoor Navigation*

Pada subbab ini akan dibahas secara mendetail dari rancangan proses sistem *indoor navigation* untuk memenuhi kebutuhan fungsionalnya. Perancangan pada sistem yang akan dibuat meliputi perancangan pemodelan lokasi dalam koordinat kartesius, perancangan pemodelan lokasi pada *graph*, perancangan proses prediksi posisi pengguna, perancangan proses mencari rute terpendek, dan perancangan kompas digital beserta panduan arah.

3.2.3.1 Perancangan Pemodelan Lokasi dalam Koordinat Kartesius

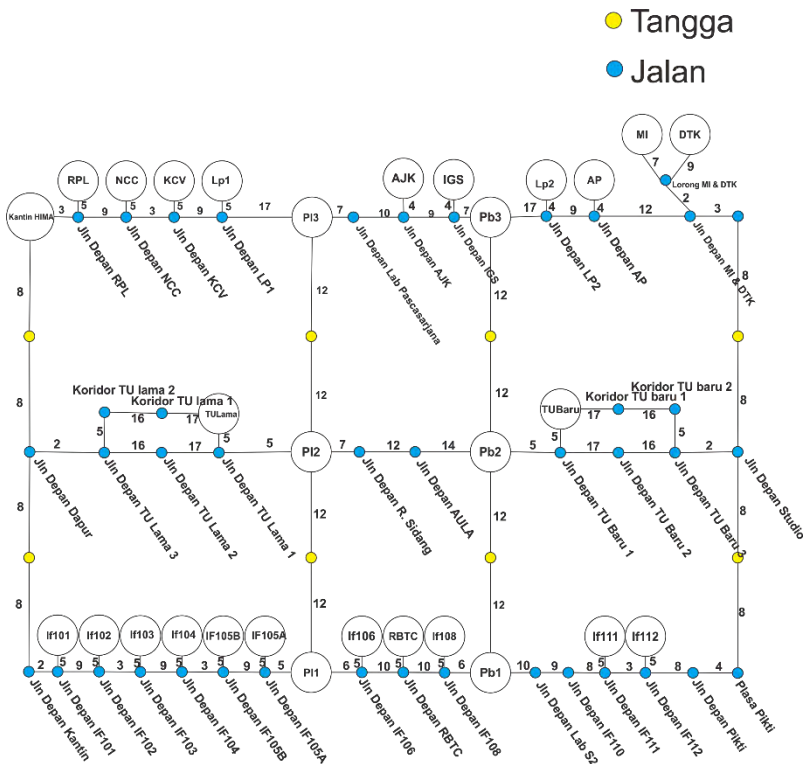
Proses ini merupakan proses untuk memodelkan gedung sebelum diimplementasikan pada program. Pertama, lokasi dimodelkan dalam koordinat kartesius (sumbu x , y dan z). Setiap ukuran 1,2 m x 1,2 m pada lokasi sebenarnya akan dimodelkan dalam satu titik koordinat. Setelah itu setiap lokasi atau ruangan diambil sebuah titik koordinat yang mewakili ruangan atau lokasi tersebut. Koordinat-koordinat ini nantinya akan digunakan pada perhitungan algoritma A^* dan implementasi panduan arah. Gambar pemberian titik koordinat ruangan atau lokasi pada lantai 1 ditunjukkan oleh Gambar 3.13. Gambar pemberian titik koordinat ruangan atau lokasi pada lantai 2 dan 3 dimuat pada bagian Lampiran A.



Gambar 3.13 Pemberian Koordinat pada Denah Gedung Lantai 1

3.2.3.2 Perancangan Pemodelan Lokasi dalam *Graph*

Setelah setiap lokasi dan ruangan diberi titik koordinat, selanjutnya setiap titik atau lokasi tersebut dihubungkan dengan garis (*edge*) sehingga menjadi sebuah *graph*. Bobot dari *edge* diambil berdasarkan pemodelan lokasi dalam koordinat kartesius. Gambar pemberian pemodelan lokasi dalam *graph* ditunjukkan pada Gambar 3.14.



3.2.3.3 Proses Prediksi Lokasi Pengguna

Proses ini merupakan proses untuk menentukan lokasi pengguna berada. Proses penentuan lokasi dilakukan dengan bantuan sistem *indoor localization*. Secara umum proses prediksi lokasi pengguna seperti berikut:

1. *Smartphone* pengguna mendapatkan data kekuatan sinyal-sinyal di posisinya berupa BSSID dan kekuatan sinyalnya.
2. Data tersebut dikirimkan ke *server* sistem *indoor localization* untuk diprediksi lokasinya.
3. *Server* menerima data RSS dari pengguna.

4. *Server* memberikan informasi lokasi ke perangkat pengguna.
5. *Smartphone* pengguna menampilkan lokasi pengguna saat ini.

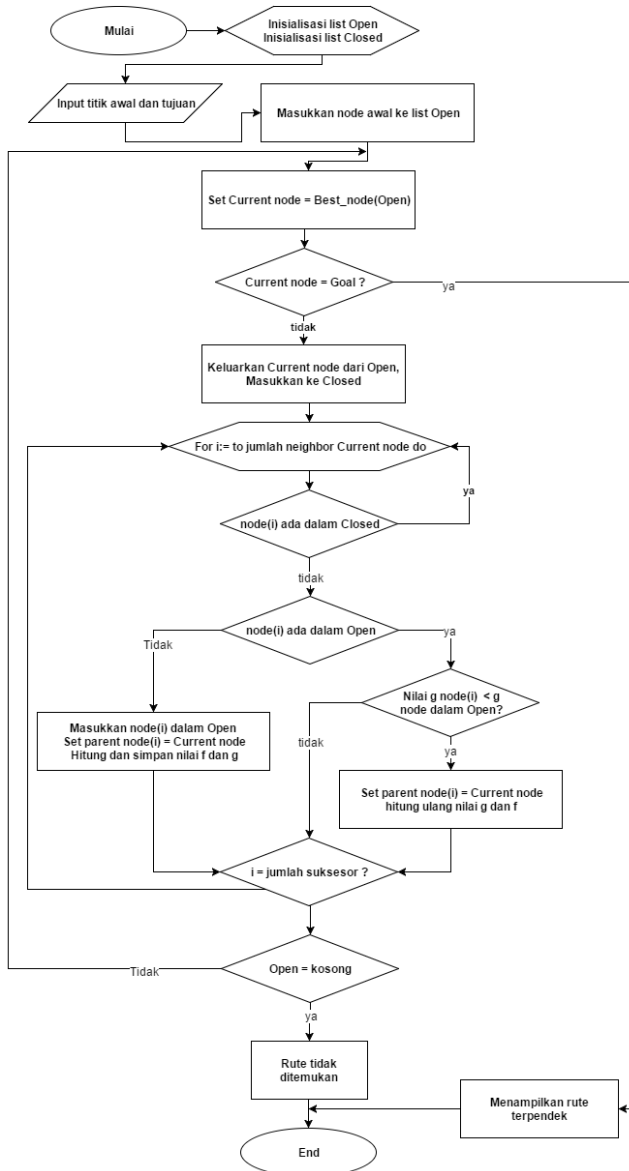
3.2.3.4 Proses Mencari Rute Terpendek

Proses ini merupakan proses mencari rute terpendek pada *graph* yang telah dibuat sebelumnya. Proses pencarian rute dilakukan setelah posisi pengguna dan tujuan diketahui. Pada Tugas Akhir ini, proses menentukan rute terpendek dilakukan dengan algoritma A*. Secara umum proses pencarian rute terpendek dengan dalam sistem *indoor navigation* seperti berikut:

1. Pengguna menentukan titik awal dan tujuan akhir dari navigasi.
2. Sistem menghitung biaya perkiraan dari *node* yang satu ke *node* yang lain dengan menggunakan *Manhattan distance*.
3. Sistem mencari rute terpendek dari *node* awal hingga tujuan dengan algoritma A*. Proses pencarian rute dengan menggunakan algoritma A* ditunjukkan oleh Gambar 3.15.
4. Rute terpendek akan ditampilkan pada peta.

3.2.3.5 Perancangan Kompas Digital dan Panduan Arah

Implementasi kompas digital dalam Tugas Akhir ini dibuat dengan memanfaatkan sensor orientasi. Sensor orientasi Android merupakan gabungan dari *geomagnetic sensor* dan *accelerometer sensor*. Salah satu nilai yang dihasilkan dari sensor tersebut adalah nilai *azimuth*. Nilai *azimuth* adalah sudut antara kutub utara magnet bumi dan sumbu y dari perangkat. Jika nilai *azimuth* sama dengan nol, maka *device* tersebut menghadap ke kutub utara magnet bumi. Nilai *azimuth* tersebut kemudian diolah dan digunakan untuk memutar Gambar 3.16 dengan bantuan animasi rotasi pada Android. Kelas yang digunakan untuk membuat animasi rotasi pada Android adalah kelas *RotateAnimation*.

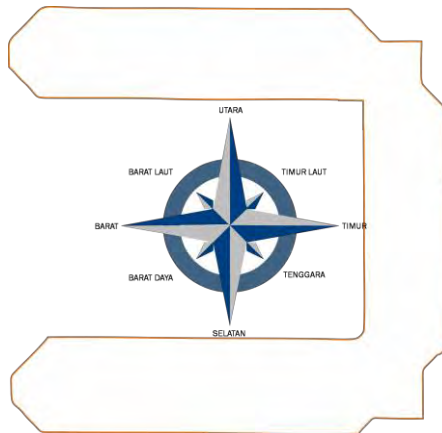


Gambar 3.15 Flowchart Algoritma A*



Gambar 3.16 Tampilan Kompas

Panduan arah yang dibuat akan menunjukkan lokasi atau ruangan selanjutnya dari posisi pengguna saat ini. Pertama, perlu dilihat posisi gedung terhadap arah mata angin. Kondisi gedung terhadap arah mata angin ditunjukkan pada Gambar 3.17. Selanjutnya, sistem dapat menentukan kemana pengguna harus pergi dengan menggunakan koordinat yang dimiliki setiap *node*.

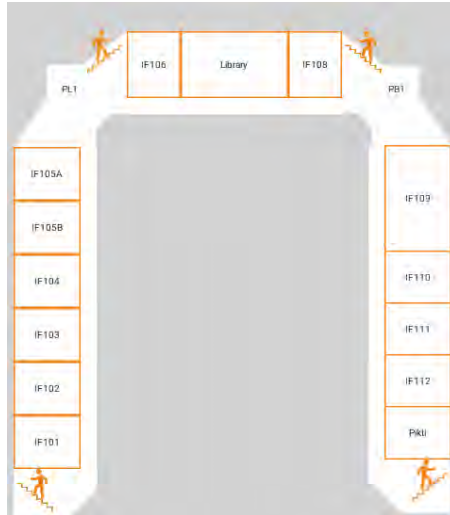


Gambar 3.17 Posisi Gedung terhadap Arah Mata Angin

3.2.3.6 Perancangan Peta Gedung Teknik Informatika ITS

Informasi tentang lokasi pengguna, tujuan pengguna, dan rute terpendek akan ditampilkan dalam sebuah peta. Gambar peta tersebut akan ditampilkan pada layar *smartphone* pengguna. Peta dibuat berdasarkan hasil pengukuran kondisi lokasi sebelumnya.

Gambar peta gedung lantai 1 ditunjukkan oleh Gambar 3.18. Gambar peta gedung lantai 2 dan 3 dimuat pada bagian Lampiran B.

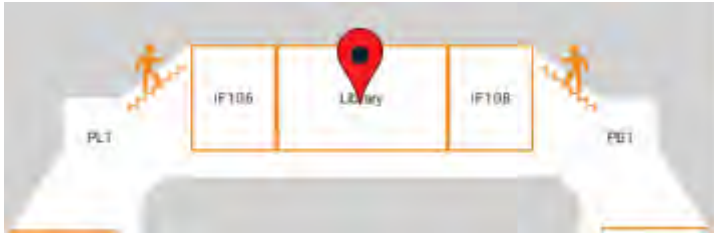


Gambar 3.18 Peta Gedung Lantai 1

Peta dibuat dalam gambar dengan format PNG (*Portable Network Graphics*). Informasi yang dimuat dalam peta yakni gambar ruangan beserta labelnya, tangga, dan jalan didepan ruang. Informasi tentang posisi pengguna berada pada peta akan ditunjukkan dengan Gambar 3.19. Contoh penggunaannya pada program ditunjukkan oleh Gambar 3.20. Sedangkan informasi tentang tujuan pengguna pada peta akan ditunjukkan dengan Gambar 3.21. Contoh penggunaannya pada program ditunjukkan pada Gambar 3.22.



Gambar 3.19 Marker Posisi Pengguna



Gambar 3.20 Contoh Penggunaan *Marker* Posisi Pengguna



Gambar 3.21 *Marker* Tujuan Pengguna



Gambar 3.22 Contoh Penggunaan *Marker* Tujuan Pengguna

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi aplikasi *indoor navigation*. Implementasi yang akan dijelaskan meliputi lingkungan implementasi baik dari sisi *hardware* maupun *software*, implementasi antarmuka, implementasi basis data, dan implementasi proses yang menunjang berjalannya sistem. Implementasi ini mengacu pada perancangan yang telah dijelaskan pada Bab 3. Akan tetapi, tidak menutup kemungkinan adanya perubahan dari rancangan apabila hal tersebut diperlukan.

4.1 Lingkungan Implementasi

Pada bagian ini dijelaskan mengenai perangkat yang digunakan pada proses implementasi aplikasi *indoor navigation* yang meliputi lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan pada proses implementasi adalah *notebook* dan *smartphone*. *Notebook* yang digunakan memiliki spesifikasi prosesor Intel® Core™ i3-3120M @2.5 GHz, 500GB HDD, dan RAM 4 GB. Sedangkan perangkat *smartphone* yang digunakan adalah *smartphone* Samsung Galaxy Grand I9082, yang memiliki spesifikasi Sistem Operasi Android v4.2 (Jelly Bean), RAM 1 GB, ROM 8 GB, prosesor Dual-core 1.2 GHz Cortex-A9, Bluetooth v4.0, A2DP, WiFi 802.11 a/b/g/n, Wi-Fi Direct, DLNA.

4.1.2 Lingkungan Implementasi Perangkat Lunak

Ada beberapa perangkat lunak yang digunakan dalam proses implementasi sistem, antara lain:

1. Microsoft Windows 8.1 sebagai sistem operasi komputer.
2. Android Studio 1.4.0 sebagai IDE untuk implementasi aplikasi.

3. SQLite sebagai sistem manajemen basis data.
4. Power Designer 15.0 untuk merancang basis data.
5. StarUML 5.0.2.1570 untuk merancang diagram-diagram perangkat lunak.
6. CorelDraw X8 untuk desain gambar.
7. Pencil untuk merancang perencanaan antarmuka aplikasi perangkat bergerak pengguna.
8. kSOAP2 untuk *library* pada *web service*.

4.2 Implementasi Basis Data

Pada subbab ini dibahas mengenai implementasi basis data, meliputi struktur basis data dan *query* terhadap basis data tersebut. Implementasi basis data ini mengacu ke perancangan basis data yang telah dirancang pada Bab 3.

4.2.1 Implementasi Struktur Basis Data

Pada bagian ini akan dibahas mengenai implementasi struktur basis data menggunakan bahasa pemrograman Java untuk membuat tabel-tabel yang telah dirancang pada Bab 3.

4.2.1.1 Implementasi Tabel Ruangan (Room)

Tabel ruangan digunakan untuk menyimpan data-data ruangan atau lokasi yang ada di sebuah bangunan. Data yang disimpan yaitu kode ruangan sebagai identitas unik ruangan dan nama ruangan. Implementasi tabel ruangan ditunjukkan Kode Sumber 4.1.

```

1. TABLE_ROOM = "Room",
2. KEY_ID_ROOM = "IdRoom",
3. KEY_ID_LOCATIONTYPE = "IdLocationType",
4. KEY_NAME = "Name";
5. db.execSQL("CREATE TABLE " + TABLE_ROOM + "(" + KEY_
  ID_ROOM + " STRING PRIMARY KEY NOT NULL," +
6. KEY_ID_LOCATIONTYPE + " STRING," +
7. KEY_NAME + " STRING)");

```

Kode Sumber 4.1 Implementasi Tabel Ruangan (Room)

4.2.1.2 Implementasi Tabel Tipe Lokasi (*LocationType*)

Tabel tipe lokasi digunakan untuk menyimpan data tipe lokasi dari sebuah ruangan atau lokasi. Data yang disimpan yaitu kode tipe ruangan sebagai identitas unik dan tipe lokasi. Implementasi tabel posisi pengguna ditunjukkan pada Kode Sumber 4.2.

```
1. TABLE_LOCATIONTYPE = "LocationType",
2. KEY_ID_LOCATIONTYPE = "IdLocationType",
3. KEY_Type = "Type";
4.
5. db.execSQL("CREATE TABLE " + TABLE_LOCATIONTYPE +
6. "(" + KEY_ID_LOCATIONTYPE + " STRING PRIMARY KEY NOT
   T NULL," + KEY_Type + " STRING)");
```

Kode Sumber 4.2 Implementasi Tabel Tipe Lokasi (*LocationType*)

4.2.1.3 Implementasi Tabel Posisi (*Position*)

Tabel posisi merupakan tabel yang digunakan untuk menyimpan semua data posisi pada gedung Teknik Informatika ITS. Data yang disimpan berupa koordinat kartesius 3 dimensi (sumbu x, y dan z) dan kode ruangan dari posisi tersebut. Implementasi tabel posisi ditunjukkan pada Kode Sumber 4.3.

```
1. TABLE_POSITION = "Position",
2. KEY_ID_POSITION = "IdPosition",
3. KEY_ID_ROOM = "IdRoom",
4. KEY_XCOOR = "XCoor",
5. KEY_YCOOR = "YCoor",
6. KEY_ZCOOR = "ZCoor";
7.
8. db.execSQL("CREATE TABLE " + TABLE_POSITION + "(" + KEY_
   ID_POSITION + " STRING PRIMARY KEY NOT NULL," +
9.     KEY_ID_ROOM + " STRING," +
10.    KEY_XCOOR + " STRING," +
11.    KEY_YCOOR + " STRING," +
12.    KEY_ZCOOR + " STRING)");
```

Kode Sumber 4.3 Implementasi Tabel Posisi (*Position*)

4.2.2 Implementasi *Query*

Query yang diimplementasikan menggunakan *Structured Query Language* (SQL). *Query* ini digunakan untuk menunjang fitur dari aplikasi *indoor navigation*. Implementasi dilakukan dengan bantuan bahasa pemrograman Java.

4.2.2.1 Implementasi *Query* Mendapatkan Daftar Tipe Lokasi

Query mendapatkan data tipe lokasi dipanggil pada saat pengguna memilih titik awal dan tujuan dari navigasi. Data yang diambil adalah data tipe lokasi beserta kode uniknya. Implementasi *query* ini ditunjukkan pada Kode Sumber 4.4.

```

1. public List<LocationType> getAllLocationType()
2. {
3.     SQLiteDatabase db = getReadableDatabase();
4.     List<LocationType> locationTypes =
5.     new ArrayList<LocationType>();
6.     Cursor cursor = db.rawQuery("SELECT * FROM " + TA
7.     BLE_LOCATIONTYPE, null);
8.     if(cursor.moveToFirst())
9.     do
10.    { LocationType locationType = new LocationType
11.    e(cursor.getString(0), cursor.getString(1));
12.    locationTypes.add(locationType);
13.    }
14.    while(cursor.moveToNext());
15.    cursor.close();
16.    db.close();
17.    return locationTypes;
18. }
```

Kode Sumber 4.4 Mendapatkan Data Tipe Lokasi

4.2.2.2 Implementasi Mendapatkan Daftar Ruang

Query mendapatkan daftar ruangan dipanggil setelah pengguna memilih tipe lokasi pada penentuan titik awal dan titik

akhir navigasi. Implementasi *query* mendapatkan daftar ruangan ditunjukkan pada Kode Sumber 4.5.

```

1. public List<Room> getRoom(String IdLocationType)
2. {
3.     SQLiteDatabase db = getReadableDatabase();
4.     List<Room> rooms = new ArrayList<Room>();
5.     Cursor cursor = db.rawQuery("SELECT * FROM " + TAB
        LE_ROOM + " WHERE " + KEY_ID_LOCATIONTYPE + "=" + I
        dLocationType , null);
6.     if(cursor==null) return null;
7.     if(cursor.moveToFirst())
8.         do
9.         {
10.             Room room = new Room(cursor.getString(0), cur
                sor.getString(1), cursor.getString(2), cursor.getStri
                ng(3), cursor.getString(4), cursor.getString(5));
11.             rooms.add(room);
12.         }
13.         while(cursor.moveToNext());
14.         cursor.close();
15.         db.close();
16.
17.         return rooms;
18. }

```

Kode Sumber 4.5 Implementasi Mendapatkan Daftar Ruangan

4.2.2.3 Implementasi *Query* Mendapatkan Data Ruangan

Query mendapatkan data ruangan ini dipanggil pada saat sistem menampilkan lokasi atau ruangan yang akan dilalui pengguna sebelum sampai tujuan akhir. Implementasi *query* ini ditunjukkan pada Kode Sumber 4.6.

```

1. public Room getDataLocation(String RoomId)
2. {
3.     SQLiteDatabase db = getReadableDatabase();
4.     Room room = null;
5.

```

```

6.   Cursor cursor = db.rawQuery("SELECT * FROM " + TAB
   LE_ROOM + " WHERE " + KEY_ID_ROOM + "=" + RoomId +
   "'" + " LIMIT 1" , null);
7.   if(cursor==null) return null;
8.
9.   if(cursor.moveToFirst())
10.  do
11.  {
12.      room = new Room(cursor.getString(0), cursor.g
   etString(1), cursor.getString(2),cursor.getString(3)
   , cursor.getString(4), cursor.getString(5));
13.  }
14.  while(cursor.moveToNext());
15.
16.      cursor.close();
17.      db.close();
18.
19.      return room;
20. }

```

Kode Sumber 4.6 Mendapatkan Data Ruangan

4.2.2.4 Implementasi *Query* Mendapatkan Data Posisi

Query mendapatkan data posisi ini dipanggil pada saat sistem menampilkan lantai dari posisi pengguna berada. Implementasi *query* ini ditunjukkan pada Kode Sumber 4.7.

```

1.  public Position getTopPosition(String RoomId) {
2.      SQLiteDatabase db = getReadableDatabase();
3.      Position position = null;
4.      Cursor cursor = db.rawQuery("SELECT * FROM " + TAB
   LE_POSITION + " WHERE " + KEY_ID_ROOM + "=" + RoomI
   d + "'" + " LIMIT 1" , null);
5.      if(cursor==null) return null;
6.      if(cursor.moveToFirst())
7.      do
8.      {
9.          position = new Position(cursor.getString(0),
   cursor.getString(1), cursor.getString(2),
10.         cursor.getString(3), cursor.getString(4));
11.      }

```

```
12.     while(cursor.moveToNext());
13.         cursor.close();
14.         db.close();
15.
16.         return position;
17. }
```

Kode Sumber 4.7 Mendapatkan Data Posisi

4.3 Implementasi Antarmuka

Pada subbab ini dibahas mengenai implementasi antarmuka yang telah dirancang pada Bab 3. Antarmuka yang dirancang adalah antarmuka pada perangkat bergerak berbasis Android.

4.3.1 Antarmuka Halaman *Splash Screen*

Halaman *splash screen* merupakan antarmuka yang pertama kali ditampilkan ketika aplikasi dijalankan. Pada antarmuka ini, pertama kali dilakukan pengecekan koneksi Wi-Fi atau koneksi data, bila tidak ditemukan maka aplikasi akan menampilkan notifikasi dan secara otomatis keluar dari aplikasi. Kemudian apabila koneksi data tersedia, akan dilakukan proses menghidupkan jalur koneksi Wi-Fi jika sebelumnya belum diaktifkan. Setelah semua proses selesai dilakukan, aplikasi akan menuju halaman informasi titik awal dan tujuan.

Pada halaman *splash screen*, terdapat sebuah gambar logo. Selain itu, terdapat *circle loading* yang berguna untuk memberi informasi kepada pengguna bahwa proses sedang berlangsung. Implementasi halaman *splash screen* ditunjukkan pada Gambar 4.1.



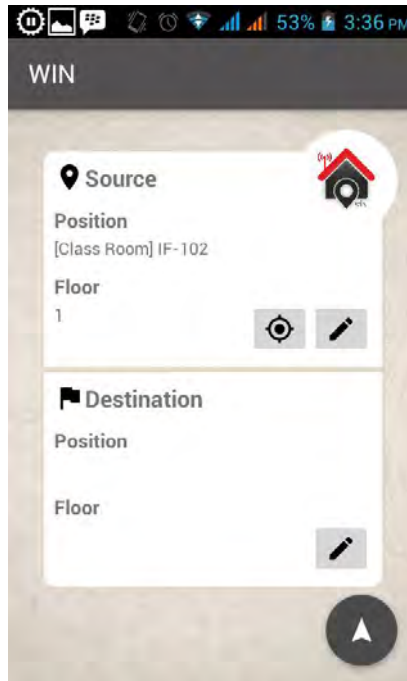
Gambar 4.1 Halaman *Splash Screen*

4.3.2 Antarmuka Halaman Informasi Titik Awal dan Tujuan

Halaman informasi titik awal (*source*) dan tujuan (*destination*) digunakan untuk menampilkan informasi tentang titik awal dan tujuan navigasi pengguna. Pada halaman ini terdapat pilihan untuk mengubah titik awal dan tujuan navigasi pengguna. Implementasi halaman ini ditunjukkan pada Gambar 4.2.

Pada halaman ini pengguna dapat mengetahui posisinya saat ini secara otomatis. Pengguna juga dapat menuju halaman *edit source* dan *edit destination* dengan tombol tersedia untuk menentukan *source* dan *destination* secara manual. Pada halaman ini pengguna juga dapat memulai navigasi. Navigasi baru dapat dimulai ketika *source* dan *destination* sudah ditentukan. Jika pengguna menekan tombol navigasi sebelum *source* dan

destination ditentukan, akan ada pemberitahuan bahwa pengguna harus menentukan dua hal tersebut terlebih dahulu.

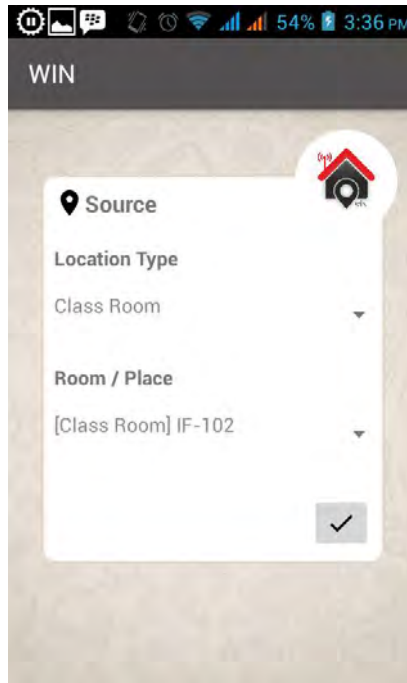


Gambar 4.2 Halaman Informasi Titik Awal dan Tujuan

4.3.3 Antarmuka Halaman *Edit Source*

Halaman *edit source* digunakan oleh pengguna untuk menentukan titik awal navigasi. Pada halaman tersebut pengguna memilih lokasi-lokasi yang tersedia sebagai titik awal. Implementasi halaman ini ditunjukkan pada Gambar 4.3.

Pada halaman ini pengguna dapat menentukan posisinya secara manual. Terdapat pilihan daftar tipe ruangan dan daftar ruangan atau lokasi. Setelah selesai menentukan posisi, pengguna akan kembali ke halaman informasi titik awal dan tujuan.

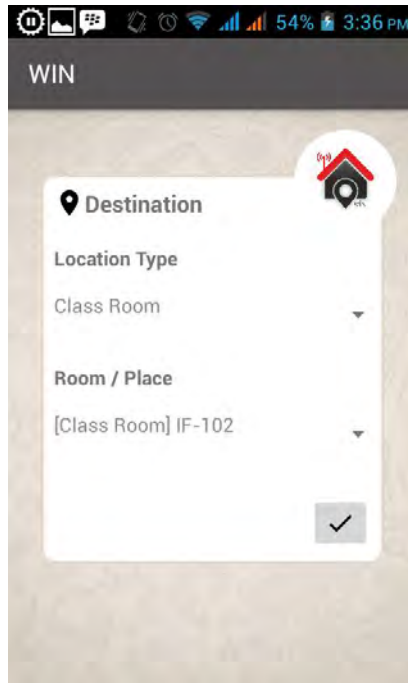


Gambar 4.3 Halaman *Edit Source*

4.3.4 Antarmuka Halaman *Edit Destination*

Halaman *edit destination* digunakan oleh pengguna untuk menentukan tujuan navigasi. Pada halaman tersebut pengguna memilih lokasi-lokasi yang tersedia sebagai tujuan. Implementasi halaman ini ditunjukkan pada Gambar 4.4.

Pada halaman ini pengguna dapat menentukan tujuan navigasinya. Terdapat pilihan daftar tipe ruangan dan daftar ruangan atau lokasi. Setelah selesai menentukan tujuan, pengguna akan kembali ke halaman informasi titik awal dan tujuan.



Gambar 4.4 Halaman *Edit Destination*

4.3.5 Antarmuka Halaman Navigasi

Halaman navigasi digunakan untuk memandu pengguna menuju tujuan. Pada halaman tersebut, terdapat peta yang menampilkan posisi pengguna, tujuan pengguna serta jalan atau rute menuju tujuan. Halaman navigasi ditunjukkan pada Gambar 4.5.

Terdapat pilihan untuk memilih gambar peta pada setiap lantai. Pengguna dapat mengetahui posisinya berada dengan tombol yang tersedia. Setelah posisi terbaru pengguna diketahui, sistem akan memperbarui rute terpendek dan panduan arah sesuai dengan posisi pengguna. Tombol *close* berguna untuk mengakhiri navigasi dan kembali ke halaman informasi titik awal dan tujuan.



Gambar 4.5 Halaman Navigasi

4.4 Implementasi Proses Sistem Aplikasi *Indoor Navigation*

Pada subbab ini akan dibahas mengenai implementasi proses yang mendukung berjalannya sistem meliputi prediksi lokasi pengguna, pencarian rute terpendek, menampilkan informasi pada peta dan kompas digital.

4.4.1 Implementasi Prediksi Lokasi Pengguna

Saat proses prediksi lokasi pengguna, aplikasi akan melakukan *scanning* sinyal WiFi untuk mendapatkan BSSID dan kekuatan sinyal yang diterima (RSS) dari beberapa *access point* yang terdeteksi di sekitar posisi pengguna. Proses ini ditunjukkan pada Kode Sumber 4.8.

```
1. private void startRefresh()
2. {
3.     wifiManager.startScan();
```

```

4. List<ScanResult> wifiScanList =
5.     wifiManager.getScanResults();
6. String rssData = "";
7. for (int i = 0; i < wifiScanList.size(); i++)
8. {
9.     rssData += wifiScanList.get(i).BSSID.replace(":", " ")
10.        + ":" + wifiScanList.get(i).level;
11.     if (i != wifiScanList.size() - 1)
12.         rssData += "#";
13.     new RefreshPosition().execute(rssData);
14. }

```

Kode Sumber 4.8 Scanning Sinyal WiFi

Selanjutnya aplikasi akan mengirimkan data hasil *scanning* sinyal Wi-Fi ke *server* sistem *indoor localization* sehingga lokasi pengguna dapat diketahui. Pengiriman data kepada *server* menggunakan menggunakan *library* kSOAP2. Fungsi yang dipanggil pada sistem *indoor localization* ketika proses prediksi lokasi adalah fungsi *GetsPosition*. Proses ini ditunjukkan pada Kode Sumber 4.9.

```

1. protected Boolean doInBackground(String...param){
2.     SoapManager soapManager =
3.         new SoapManager(UserInput.this);
4.     data = soapManager.GetsPosition(param[0]);
5.     if(!data.contains("ERROR")) return true;
6.     else return false;
7. }

```

Kode Sumber 4.9 Pengiriman Hasil Scanning Sinyal WiFi

4.4.2 Implementasi Pencarian Rute Terpendek

Proses pencarian rute terpendek dilakukan setelah lokasi dan tujuan pengguna diketahui. Seperti yang sudah dijelaskan sebelumnya, algoritma yang digunakan untuk mencari rute terpendek adalah algoritma A*. Algoritma ini memerlukan

perhitungan perkiraan *cost* dari sebuah *node* ke *node* yang lain atau atau biaya heuristik.

Fungsi heuristik pada Tugas Akhir ini menggunakan fungsi *Manhattan distance*. Biaya heuristik yang dihitung dari titik koordinat x, y, dan z yang dimiliki setiap *node*. Implementasi perhitungan biaya heuristik ditunjukkan pada Kode Sumber 4.10.

```

1. for(int a=0; a<cells.size(); a++)
2. {
3.     ...
4.     for(int b=0; b<cells.size(); b++)
5.     {
6.         map.put(cells.get(b).getNodeId(), Math.abs(cells
           .get(a).getX()-
           cells.get(b).getX()+Math.abs(cells.get(a).getY()-
           cells.get(b).getY()+Math.abs(cells.get(a).getZ()-
           cells.get(b).getZ()));
7.     }
8.     ...
9. }

```

Kode Sumber 4.10 Implementasi Fungsi *Manhattan Distance*

Setelah biaya heuristik antar *node* dihitung, selanjutnya adalah implementasi algoritma A*. Dalam mencari rute terpendek, algoritma ini menggunakan dua buah penyimpanan *node*, yakni *openQueue* dan *closedList*. *Node* yang berada di *openQueue* merupakan *node* yang pernah diperiksa dan nilai heuristiknya telah dihitung, tapi belum terpilih sebagai *best node*. Sebuah *node* akan disimpan di *closedList* jika *node* tersebut pernah terpilih sebagai *best node*. Dengan kata lain *closedList* berisi *node* yang tidak mungkin terpilih sebagai simpul terbaik. *Pseudocode* implementasi algoritma A* ditunjukkan pada Kode Sumber 4.12.

Setelah tujuan ditemukan, maka sistem akan memanggil fungsi untuk merekonstruksi *path* dari *source* sampai *destination*. Implementasi dari fungsi ini ditunjukkan pada Kode Sumber 4.11.

```

1. private List<T> path(Map<T, T> path, T destination){
2.
3.     final List<T> pathList = new ArrayList<T>();
4.     pathList.add(destination);
5.     while (path.containsKey(destination)) {
6.         destination = path.get(destination);
7.         pathList.add(destination);
8.     }
9.     Collections.reverse(pathList);
10.    return pathList;
11. }

```

Kode Sumber 4.11 Fungsi Merekonstruksi *Path*

```

1. function astar(source, destination) {
2.     openQueue ← new PriorityQueue
3.     sourceNodeData ← NodeData(source)
4.     sourceNodeData.setG(0)
5.     sourceNodeData.calcF(destination)
6.     openQueue.add(sourceNodeData)
7.
8.     path ← new HashMap
9.     closedList ← new HashSet
10.
11.     while openQueue is not Empty
12.         nodeData ← openQueue.poll
13.         if nodeData equals destination
14.             return path
15.         Endif
16.         closedList.add(nodeData)
17.
18.         for each neighbor of nodeData
19.             if closedList contains neighbour
20.                 continue
21.             Endif
22.
23.             distanceBetweenTwoNodes ← neighbor.getValue
24.             tentativeG ←
25.                 distanceBetweenTwoNodes + nodeData.getG
26.
27.             if tentativeG < neighbor.getG

```

```

28.     neighbor.setG(tentativeG)
29.     neighbor.calcF(destination)
30.
31.     path.put(neighbor, nodeData)
32.     if openQueue not contains neighbor
33.         openQueue.add(neighbor)
34.     Endif
35. Endif
36. Endfor
37. return null;
38. }

```

Kode Sumber 4.12 Pseudocode Implementasi Algoritma A*

4.4.3 Implementasi Menampilkan Informasi pada Peta

Pada subbab ini dibahas mengenai implementasi menampilkan peta yang meliputi implementasi merubah warna jalan pada peta, menempatkan *marker* posisi pengguna, dan menempatkan *marker* tujuan pengguna.

4.4.3.1 Implementasi Merubah Warna Jalan pada Peta

Rute terpendek yang sudah ditentukan akan ditampilkan pada peta dengan warna yang berbeda. Hal tersebut dilakukan dengan cara merubah warna gambar jalan atau ruangan yang akan dilewati pengguna. Implementasi dari fungsi ini ditunjukkan pada Kode Sumber 4.13.

```

1. private void changeRoadColor(String nodeId) {
2.     int imageViewId = getResources().getIdentifier(no
   deId, "id", "com.example.alifsip.win");
3.     ImageView imageView =
4.         (ImageView) findViewById(imageViewId);
5.     imageView.setColorFilter(Color.argb(255, 255, 255
   , 153));
6. }

```

Kode Sumber 4.13 Implementasi Merubah Warna Jalan pada Peta

4.4.3.2 Implementasi Menempatkan *Marker* Posisi Pengguna

Posisi pengguna pada peta ditunjukkan oleh sebuah gambar *marker*. Setiap pengguna memperbarui posisinya, gambar *marker* akan disesuaikan posisinya dengan posisi pengguna pada peta. Implementasi dari fungsi ini ditunjukkan pada Kode Sumber 4.14.

```

1. private void setSourceMarker() {
2.     int imageSourceId = getResources().getIdentifier(s
       source, "id", "com.example.alifsip.win");
3.     ImageView imageViewSource =
4.     (ImageView) findViewById(imageSourceId);
5.
6.     RelativeLayout viewStart;
7.     viewStart = (RelativeLayout)
       findViewById(R.id.pointerStart);
8.
9.     Rect loc = new Rect();
10.    int[] location = new int[2];
11.    imageViewSource.getLocationOnScreen(location);
12.
13.    loc.left = location[0];
14.    loc.top = location[1];
15.    loc.right =
16.    loc.left + imageViewSource.getWidth();
17.    loc.bottom =
18.    loc.top + imageViewSource.getHeight();
19.
20.    viewStart.setPadding(loc.left+((loc.right-
       loc.left)/2),loc.top,0,0);
21. }

```

Kode Sumber 4.14 Implementasi Menempatkan *Marker* Posisi Pengguna

4.4.3.3 Implementasi Menempatkan *Marker* Tujuan Pengguna

Tujuan pengguna pada peta ditunjukkan oleh sebuah gambar *marker*. Gambar *marker* tujuan akan disesuaikan setelah pengguna

memulai sebuah sesi navigasi. Implementasi dari fungsi ini ditunjukkan pada Kode Sumber 4.15.

```

1. private void setDesMarker() {
2.     int imageDesId = getResources().getIdentifier(dest
        ination, "id", "com.example.alifsip.win");
3.     ImageView imageViewDes =
4.         (ImageView) findViewById(imageDesId);
5.
6.     RelativeLayout viewFinish;
7.     viewFinish = (RelativeLayout)
        findViewById(R.id.finishFlag);
8.
9.     Rect loc = new Rect();
10.    int[] location = new int[2];
11.    imageViewDes.getLocationOnScreen(location);
12.
13.    loc.left = location[0];
14.    loc.top = location[1];
15.    loc.right =
16.        loc.left + imageViewDes.getWidth();
17.    loc.bottom =
18.        loc.top + imageViewDes.getHeight();
19.
20.    viewFinish.setPadding(loc.left+((loc.right-
        loc.left)/2),loc.top,0,0);
21. }

```

Kode Sumber 4.15 Implementasi Menempatkan *Marker* Tujuan Pengguna

4.4.4 Implementasi Kompas Digital dan Panduan Arah

Implementasi kompas digital dimulai dengan mengaktifkan sensor orientasi pada perangkat Android. Sensor tersebut akan menghasilkan nilai *azimuth* yang digunakan pada proses selanjutnya. Implementasi dari fungsi ini ditunjukkan pada Kode Sumber 4.16.

```

1. @Override
2. protected void onResume() {

```

```

3.  super.onResume();
4.
5.  mSensorManager.registerListener(this,
6.  mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTAT
    ION), SensorManager.SENSOR_DELAY_GAME);
7.  }

```

Kode Sumber 4.16 Aktifasi Sensor Orientasi

Setelah sensor orientasi diaktifkan, nilai *azimuth* yang didapat digunakan untuk memutar gambar tampilan kompas dengan bantuan animasi Android. Kelas yang digunakan untuk membuat animasi rotasi pada Android yakni kelas *RotateAnimation*. Implementasi animasi rotasi pada gambar tampilan kompas ditunjukkan pada Kode Sumber 4.18.

Implementasi panduan arah dimulai dengan menentukan koordinat *node* pengguna berada saat ini (*current node*) dan koordinat *node* selanjutnya (*next node*). Koordinat tersebut berguna untuk menentukan arah kemana pengguna harus pergi. Implementasi dari panduan arah ditunjukkan pada Kode Sumber 4.17.

```

1.  private void setNextNode()
2.  {
3.      ...
4.      for(Cell c : cells){
5.          if(c.getNodeId() != null &&
6.             c.getNodeId().contains(source))
7.          {
8.              currentX = c.getX();
9.              currentY = c.getY();
10.         }
11.         if (c.getNodeId() != null &&
12.            c.getNodeId().contains(nextNode))
13.         {
14.             nextX = c.getX();
15.             nextY = c.getY();
16.         }
17.     }

```

```

18.
19.  if (currentX == nextX)
20.  {
21.      if(currentY < nextY)
22.          txt_suggestion.setText("Go to the east");
23.      else
24.          txt_suggestion.setText("Go to the west");
25.  }
26.  else if (currentY == nextY)
27.  {
28.      if(currentX < nextX)
29.          txt_suggestion.setText("Go to the south")
30.      else
31.          txt_suggestion.setText("Go to the north");
32.  }
33. }

```

Kode Sumber 4.17 Implementasi Panduan Arah

```

1. @Override
2. public void onSensorChanged(SensorEvent sensorEvent)
3. {
4.     float degree = Math.round(sensorEvent.values[0]);
5.
6.     RotateAnimation ra = new RotateAnimation(
7.         currentDegree,
8.         -degree,
9.         Animation.RELATIVE_TO_SELF, 0.5f,
10.        Animation.RELATIVE_TO_SELF, 0.5f);
11.
12.        ra.setDuration(210);
13.        ra.setFillAfter(true);
14.        arrow.startAnimation(ra);
15.        currentDegree = -degree;
16.    }

```

Kode Sumber 4.18 Animasi Rotasi pada Gambar Tampilan Kompas

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji fungsionalitas dan uji akurasi serta analisa setiap pengujian.

5.1 Lingkungan Uji Coba

Lingkungan pengujian merupakan perangkat-perangkat yang dilibatkan dalam proses pengujian. Lingkungan pengujian ini menggunakan perangkat keras berupa perangkat bergerak berbasis Android yang terhubung dengan internet, memiliki fasilitas WLAN, dan dilengkapi dengan sensor *magnetometer*. Spesifikasi lingkungan pengujian dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Pengujian Perangkat Lunak

Nama Smartphone	Samsung Galaxy Grand I9082
Sistem Operasi	Android v4.2 (Jelly Bean)
Prosesor	Dual-core 1.2 GHz Cortex-A9
RAM	1 GB
WLAN	Wi-Fi 802.11 a/b/g/n
Network Speed	HSPA 21.1/5.76 Mbps

5.2 Dasar Pengujian

Pengujian yang dilakukan berupa pengujian fungsionalitas dan pengujian akurasi sistem. Pengujian fungsionalitas dilakukan dengan model *black box* untuk masing-masing fungsionalitas dari aplikasi ini. Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasikan dan bekerja seperti yang diharapkan. Pengujian akurasi sistem dilakukan dengan cara menggunakan

aplikasi untuk melakukan navigasi dengan skenario yang telah dibuat sebelumnya.

5.3 Pengujian Fungsionalitas

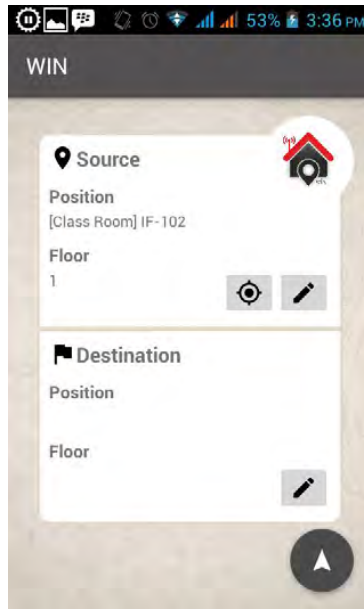
Pada subbab ini dibahas pengujian yang dilakukan pada aplikasi *indoor navigation*. Pengujian ini dilakukan dengan tujuan untuk memastikan fungsionalitas pada perangkat telah berjalan dengan benar.

5.3.1 Pengujian Menampilkan Posisi Pengguna

Berikut ini merupakan pembahasan pengujian untuk menampilkan posisi pengguna. Pada proses mengetahui posisi pengguna, dibutuhkan data sinyal Wi-Fi. Oleh karena itu, pengguna diharuskan untuk memastikan bahwa perangkat Wi-Fi pada perangkat bergerakaknya telah menyala. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.2.

Tabel 5.2 Skenario Pengujian Mengetahui Lokasi Pengguna

Nomor	UT-01
Nama	Menampilkan Posisi Pengguna
Kasus Penggunaan	UC-01
Tujuan	Mengetahui posisi pengguna berada
Kondisi awal	Posisi pengguna belum diketahui
skenario	1. Pengguna membuka aplikasi
	2. Pengguna menekan tombol <i>my location</i>
masukan	-
Keluaran yang diharapkan	Nama lokasi pengguna diketahui dan ditampilkan
Hasil pengujian	Berhasil



Gambar 5.1 Pengujian Mengetahui Lokasi Pengguna

5.3.2 Pengujian Melakukan Navigasi

Berikut ini merupakan pembahasan pengujian untuk melakukan navigasi. Pada proses navigasi, pengguna menentukan titik awal dan tujuan dari navigasi, selanjutnya sistem akan mencari rute terpendek dari titik awal dan tujuan pengguna yang telah ditentukan sebelumnya. Selain rute, sistem juga menampilkan penunjuk arah sebagai alat bantu navigasi. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.3.

Tabel 5.3 Skenario Pengujian Melakukan Navigasi

Nomor	UT-02
Nama	Melakukan Navigasi
Kasus Penggunaan	UC-02

Tujuan	Menampilkan rute menuju tujuan serta panduan arah
Kondisi awal	Rute dan panduan arah belum ditampilkan
Skenario 1	1. Pengguna menentukan titik awal
	2. Pengguna menentukan titik akhir
	3. Pengguna menekan tombol navigasi
Masukan	-
Keluaran yang diharapkan	Rute terpendek dan panduan arah ditampilkan
Hasil pengujian	Berhasil
Skenario 2	1. Pengguna menentukan titik awal
	2. Pengguna menentukan titik akhir
	3. Pengguna menekan tombol navigasi
	4. Pengguna menekan tombol <i>update</i>
Masukan	-
Keluaran yang diharapkan	Rute terpendek dan panduan arah diperbarui sesuai posisi pengguna saat ini
Hasil pengujian	Berhasil
Skenario 3	1. Pengguna menentukan titik awal
	2. Pengguna menentukan titik akhir
	3. Pengguna menekan tombol navigasi
	4. Pengguna menekan tombol pilihan lantai
Masukan	-
Keluaran yang diharapkan	Tampilan peta sesuai dengan pilihan lantai pengguna
Hasil pengujian	Berhasil

5.4 Pengujian Akurasi

Pengujian akurasi dilakukan dengan cara menggunakan aplikasi untuk melakukan navigasi pada keadaan yang sebenarnya. Pengujian ini bertujuan untuk mengukur tingkat akurasi sistem dalam membantu penggunaannya saat menentukan posisi pengguna berada serta mencari rute terpendek menuju tujuan.



Gambar 5.2 Pengujian Melakukan Navigasi

5.4.1 Skenario Pengujian Akurasi

Pada bagian ini akan dilakukan uji coba terhadap keakuratan aplikasi ketika digunakan untuk melakukan navigasi. Uji coba ini dilakukan dengan menggunakan 6 kasus percobaan. Setiap kasus percobaan memiliki titik awal (*source*) dan titik akhir (*destination*) yang berbeda. Kasus-kasus percobaan dalam pengujian akurasi ini terdapat pada Tabel 5.4.

Tabel 5.4 Kasus Pengujian Akurasi

Nama Kasus	Titik Awal	Titik Akhir
Kasus 1	Jalan Depan IF105A	Ruang Administrasi Lama
Kasus 2	Lab. Algoritma Pemrograman	Lab. Pemrograman 1

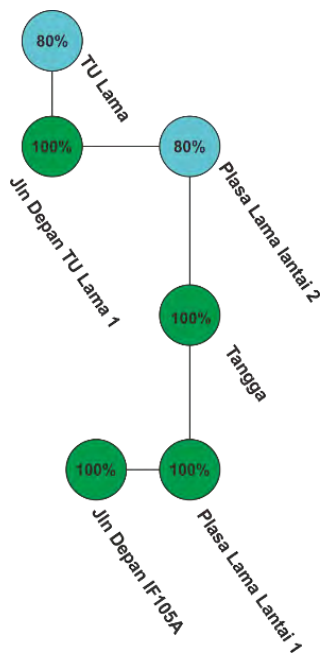
Kasus 3	Ruang Administrasi Lama	Ruang Administrasi Baru
Kasus 4	Plasa Baru lantai 1	Lab. Arsitektur dan Jaringan
Kasus 5	Plasa Baru lantai 1	Lab. Rekayasa Perangkat Lunak
Kasus 6	Lab. Algoritma Pemrograman	Plasa Lama lantai 1

Setiap kasus percobaan navigasi akan dilakukan sebanyak 5 kali. Setiap dilakukan percobaan, dilakukan prediksi posisi pengguna dan pencarian rute terpendek pada saat pengguna melewati ruangan atau lokasi yang dilalui.

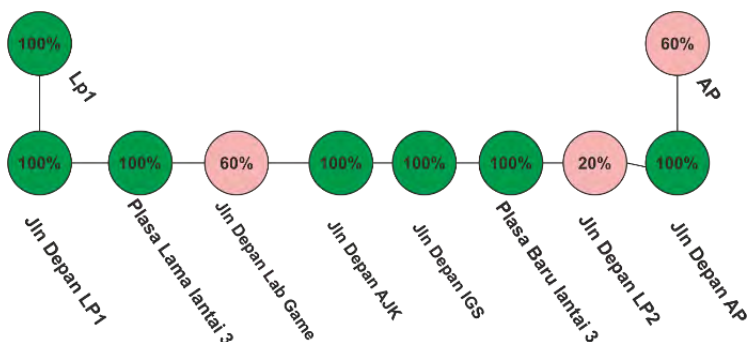
5.4.2 Hasil Pengujian Akurasi

Nilai akurasi yang dihitung pada pengujian ini adalah akurasi penentuan posisi pengguna dan pencarian rute terpendek. Nilai akurasi pendeteksian posisi pengguna dihitung dengan mencari rata-rata dari lima kali percobaan pada setiap *node* yang dilalui ketika melakukan navigasi. Hasil pengujian pada kasus 1 dan kasus 2 ditunjukkan oleh Gambar 5.3 dan Gambar 5.4. Hasil pengujian pada kasus 3, kasus 4, kasus 5, dan kasus 6 dimuat pada bagian Lampiran C.

Gambar hasil pengujian menunjukkan rute terpendek dan akurasi pendeteksian posisi pengguna pada setiap kasus percobaan. Warna pada *node* menunjukkan tingkat akurasi. Warna hijau menandakan tingkat akurasi sempurna, warna biru menandakan tingkat akurasi 80%, warna merah menandakan tingkat akurasi kurang dari 80%.



Gambar 5.3 Hasil Pengujian pada Kasus 1



Gambar 5.4 Hasil Pengujian pada Kasus 2

Secara keseluruhan, sistem *indoor localization* yang digunakan menghasilkan akurasi sebesar 88,953% pada seluruh

node yang dilalui. Terdapat 43 *node* yang memiliki akurasi 100%, 8 *node* dengan akurasi 80%, 7 *node* dengan akurasi 60%, 2 *node* dengan akurasi 40%, dan 2 *node* dengan akurasi 20%.

5.4.3 Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian fungsionalitas dan pengujian akurasi, maka dapat dilakukan evaluasi sebagai berikut.

5.4.3.1 Evaluasi Pengujian Fungsionalitas

Pengujian fungsionalitas yang telah dilakukan memberikan hasil yang sesuai dengan skenario yang telah direncanakan. Evaluasi pengujian pada masing-masing fungsionalitas dijelaskan sebagai berikut.

1. Pengujian mengetahui lokasi pengguna telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UT-01 yang memberikan informasi bahwa proses mengetahui lokasi pengguna telah berjalan dengan baik.
2. Pengujian melakukan navigasi telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UT-02 yang memberikan informasi bahwa proses melakukan navigasi telah berjalan dengan baik.

5.4.3.2 Evaluasi Pengujian Akurasi

Pengujian yang dilakukan terhadap sistem *indoor navigation* dilakukan dengan menguji keakuratan hasil pendeteksian lokasi pengguna dan pencarian rute yang dihasilkan. Secara umum, sistem *indoor localization* yang digunakan untuk menggantikan fungsi GPS dalam menentukan posisi pengguna memberikan performa yang baik. Sistem tersebut mampu menghasilkan presentase rata-rata akurasi pendeteksian lokasi pengguna sebesar 88,953% diambil dari seluruh kasus pada setiap lokasi yang dilewati dalam percobaan. Hal itu jelas lebih baik dibandingkan

menggunakan GPS yang tidak mampu memberikan informasi posisi pengguna secara akurat didalam gedung bertingkat.

Sedangkan untuk uji coba pencarian rute, sistem *indoor navigation* dapat memberikan rute terpendek pada semua kasus percobaan yang dibuat. Hal ini menunjukkan bahwa modul pencarian rute terpendek yang memanfaatkan algoritma A* telah diimplementasikan dengan baik.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan kesimpulan yang dapat diambil dalam pengerjaan Tugas Akhir dan saran untuk pengembangan lebih lanjut dari sistem *indoor navigation* pada perangkat bergerak berbasis Android.

6.1 Kesimpulan

Dalam proses pengerjaan Tugas Akhir dari tahap pendahuluan, kajian pustaka, analisis, perancangan, implementasi dan pengujian sistem *indoor navigation*, diperoleh kesimpulan sebagai berikut.

1. Proses memanfaatkan sistem *indoor localization* menggunakan sinyal Wi-Fi sebagai penentu posisi pengguna pada sistem *indoor navigation* dilakukan dengan memanggil fungsi *GetPosition* pada *web service*.
2. Sistem *indoor navigation* mencari rute terpendek dengan mengimplementasikan algoritma A* menggunakan bahasa pemrograman Java pada perangkat Android.
3. Implementasi kompas digital pada sistem ini dilakukan dengan cara memanfaatkan nilai *azimuth* yang diperoleh dari sensor orientasi pada perangkat Android.

6.2 Saran

Berikut ini merupakan beberapa saran mengenai pengembangan lebih lanjut aplikasi *indoor navigation* berdasarkan hasil rancangan, implementasi dan uji coba yang telah dilakukan.

1. Menggunakan sistem *realtime indoor localization* untuk menentukan posisi pengguna.
2. Menerapkan animasi pergerakan posisi pengguna.
3. Menambahkan panduan arah berupa suara.
4. Menggabungkan GPS dan *indoor localization* untuk membuat sistem navigasi yang dapat digunakan pada kasus *indoor* dan *outdoor*.

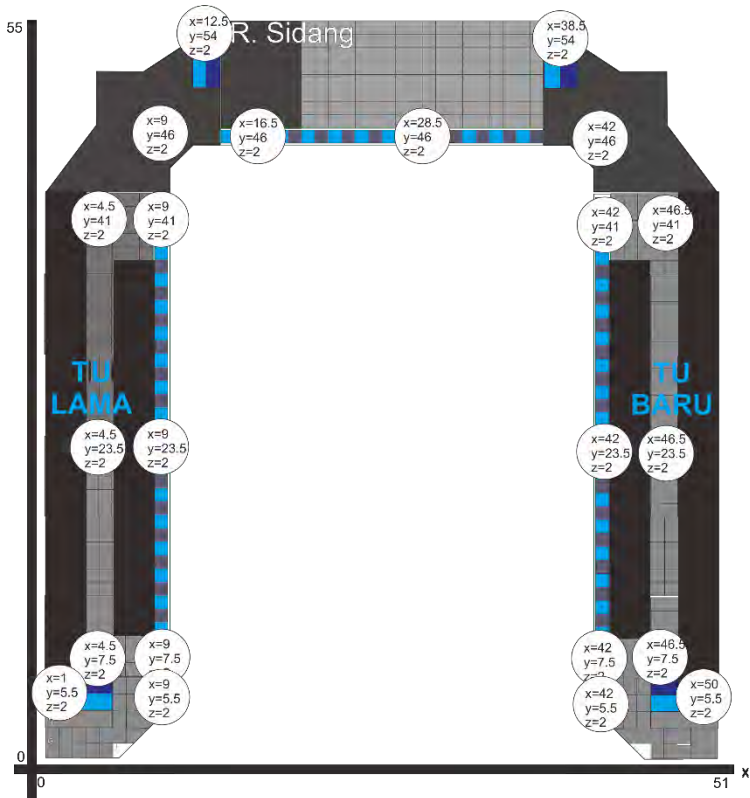
[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

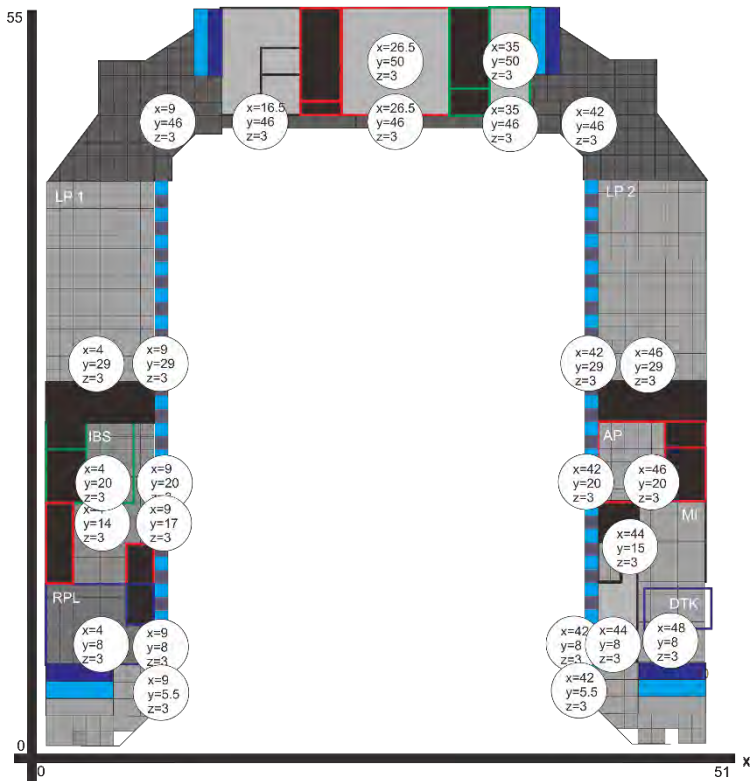
- [1] H. Fredrick, "Why Doesn't GPS Work Inside a Building?," OpposingViews.com, [Online]. Available: <http://science.opposingviews.com/doesnt-gps-work-inside-building-18659.html>. [Diakses 25 Mei 2016].
- [2] Nick, "How to use an Android phone as a compass," [Online]. Available: http://www.phonearena.com/news/How-to-use-an-Android-phone-as-a-compass_id61926. [Diakses 14 Juni 2015].
- [3] M. F. Ghanianto, Implementasi Indoor Localization Menggunakan Sinyal Wifid dan Clustering Filtered K-Nearest Neighbors untuk Pelacakan Keberadaan Seseorang dan Evaluasi Akurasi Pelacakan di Kampus Teknik Informatika ITS, Surabaya, 2015.
- [4] S. Steiniger, M. Neun dan A. Edwardes, "Foundations of Location Based Services," *CartouCHE-Cartography for Swiss Higher Education*, vol. 1.0, p. 2, 2008.
- [5] N. Y. Arrifa, R. V. H. Ginardi dan A. M. Shiddiqi, "Implementasi Indoor Localization Menggunakan Sinyal Wi-Fi dan Decision Tree untuk Pelacakan Keberadaan Seseorang di Kampus Teknik Informatika ITS," *Jurnal Teknik POMITS*, vol. 2, pp. 33-40, 2014.
- [6] "Gedung dan Ruang Baca," [Online]. Available: <http://if.its.ac.id/fasilitas/gedung-dan-ruang-baca/>. [Diakses 22 Juni 2016].
- [7] "Android Studio The Official IDE For Android," [Online]. Available: <http://developer.android.com/sdk/index.html>. [Diakses 9 Mei 2016].
- [8] "About SQLite," [Online]. Available: <https://www.sqlite.org/about.html>. [Diakses 11 Juni 2016].

- [9] Pawitri, “Implementasi Algoritma PHYSICAL-A* (PHA*) untuk menemukan Lintasan Terpendek,” dalam *Seminar Nasional Teknologi 2007*, Yogyakarta, 2007.
- [10] R. Munir, *Matematika Diskrit*. Edisi Ketiga, Bandung: Informatika, 2005.
- [11] R. Munir, *Matematika Diskrit*. Edisi Revisi Kelima, Bandung: Informatika, 2012.
- [12] S. Russel dan P. Norvig, *Artificial Intelligence: A Modern Approach*, New Jersey: Pearson Education, Inc, 2010.
- [13] “Position Sensors,” [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_position.html#sensors-pos-orient. [Diakses 21 Mei 2016].

LAMPIRAN A - Pemodelan Lokasi dalam Koordinat Kartesius

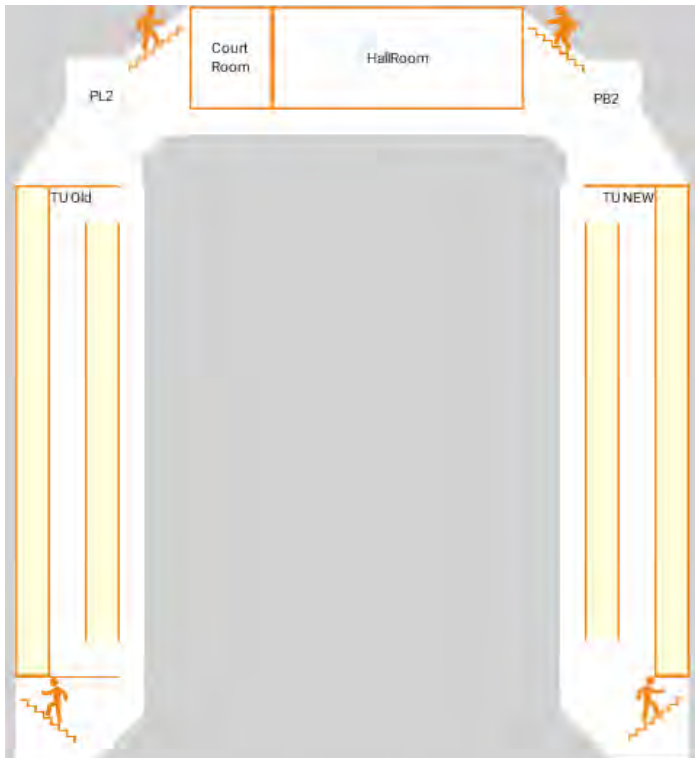


Gambar 7.1 Pemodelan Lokasi dalam Koordinat Kartesius Lantai 2

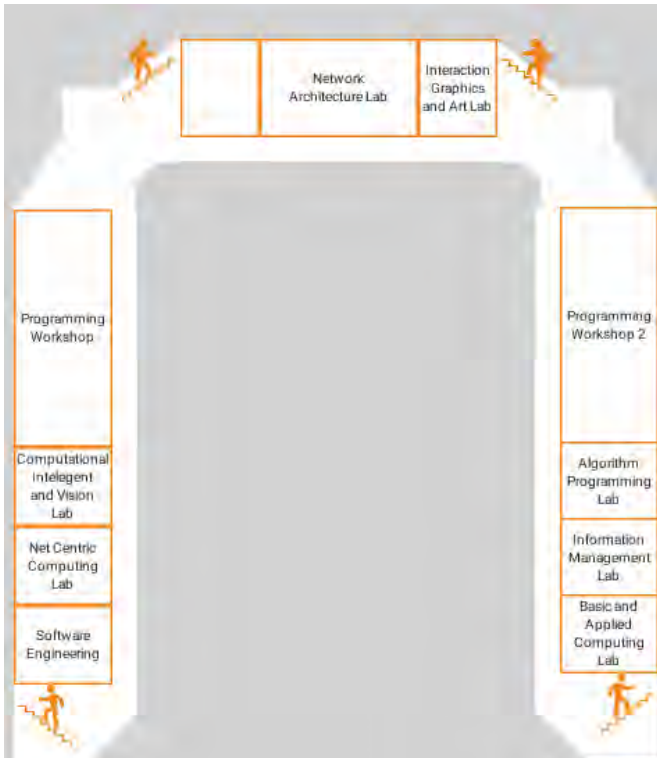


Gambar 7.2 Pemodelan Lokasi dalam Koordinat Kartesius Lantai 3

LAMPIRAN B – Gambar Peta



Gambar 8.1 Gambar Peta Lantai 2

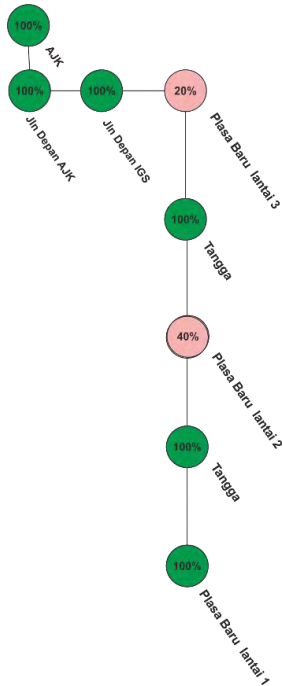


Gambar 8.2 Gambar Peta Lantai 3

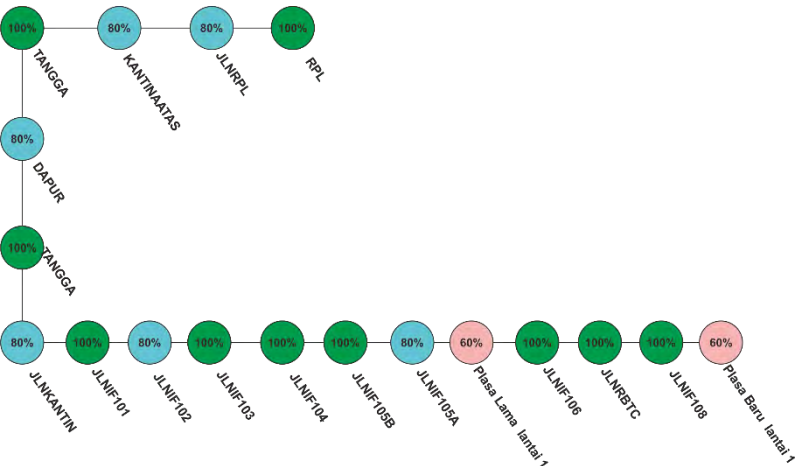
LAMPIRAN C – Gambar Hasil Pengujian



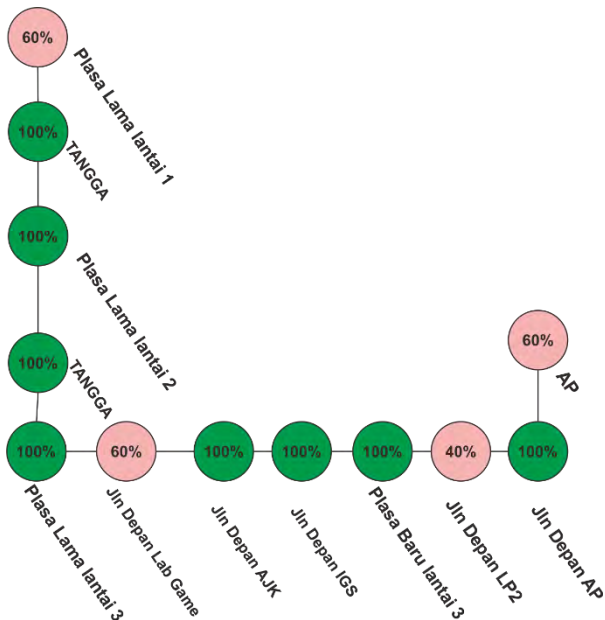
Gambar 9.1 Hasil Pengujian pada Kasus 3



Gambar 9.2 Hasil Pengujian pada Kasus 4



Gambar 9.3 Hasil Pengujian pada Kasus 5



Gambar 9.4 Hasil Pengujian pada Kasus 6

BIODATA PENULIS



Alifa Ridho Musthafa, biasa dipanggil Alif, dilahirkan di kota Klaten pada tanggal 8 November 1993. Penulis merupakan putra pertama dari dua bersaudara yang besar di Klaten. Penulis menempuh pendidikan pra sekolah di TK NU Pembangunan Klaten, selanjutnya melanjutkan pendidikan formalnya SDN 3 Puluhan, SMP Negeri 1 Pedan, dan SMA Negeri 1

Karanganom jurusan Ilmu Pengetahuan Alam. Penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Dalam menyelesaikan pendidikan S1, penulis mengambil rumpun mata kuliah Algoritma Pemrograman.

Pada dunia non akademik, penulis aktif di organisasi kemahasiswaan jurusan HMTK (Himpunan Mahasiswa Teknik Computer-Informatika) sebagai staf departemen kesejahteraan mahasiswa pada tahun 2013-2014 dan staf ahli departemen kesejahteraan mahasiswa pada tahun 2014-2015. Penulis juga pernah menjadi anggota sie keamanan dan perijinan pada kegiatan SCHEMATICS 2013 dan menjadi staf kesekretariatan pada SCHEMATICS 2014. Selain itu, penulis juga pernah menjadi staf ahli Bina Muslim Informatika organisasi jurusan KMI (Keluarga Muslim Informatika) pada tahun 2014-2015.